

日 本 国 特 許 庁
JAPAN PATENT OFFICE

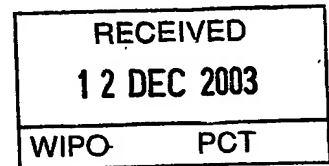
23.10.03

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日
Date of Application: 2 0 0 2 年 1 0 月 2 4 日

出 願 番 号
Application Number: 特 願 2 0 0 2 - 3 0 9 6 8 0
[ST. 10/C]: [J P 2 0 0 2 - 3 0 9 6 8 0]



出 願 人
Applicant(s): 株式会社ターボデータラボラトリー

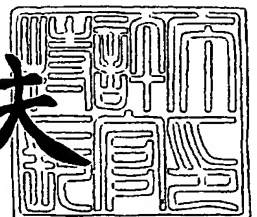
PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH
RULE 17.1(a) OR (b)

BEST AVAILABLE COPY

2 0 0 3 年 1 1 月 2 7 日

特許庁長官
Commissioner,
Japan Patent Office

今 井 康 夫



【書類名】 特許願

【整理番号】 PK010107

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 17/30

【発明者】

【住所又は居所】 神奈川県横浜市神奈川区松見町四丁目 1 1 0 1 番地 7 コ
ートハウス菊名 8 0 4 号

【氏名】 古庄 晋二

【特許出願人】

【識別番号】 502369012

【氏名又は名称】 株式会社ターボデータラボラトリー

【代理人】

【識別番号】 100103632

【弁理士】

【氏名又は名称】 窪田 英一郎

【選任した代理人】

【識別番号】 100099715

【弁理士】

【氏名又は名称】 吉田 聡

【手数料の表示】

【予納台帳番号】 058377

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【ブルーフの要否】 要

【書類名】 明細書

【発明の名称】 連鎖したジョインテーブルのツリー構造への変換方法、および、変換プログラム

【特許請求の範囲】

【請求項1】 各々が、項目とこれに含まれる項目値とを含むレコードの配列として表わされる複数の表形式データを結合し、結合された表形式データであるジョインテーブルをツリー構造に変換する方法であって、

a) 各表形式データを、各々が、特定の項目に属する項目値に対応した項目値番号の順に当該項目値が格納されている値リストと、一意的なレコード番号の順に、当該項目値番号を指示するためのポインタ値が格納されたポインタ配列とからなる一以上の情報ブロックに分割するように構成するステップと、

b) 項目を共通化すべき二つの表形式データを選択するステップと、

c) 当該選択された二つの表形式データにおいて、共通化すべき項目を見出すステップと、

d) 前記特定された情報ブロックのうち、デフォルトのソート順が反映される表形式データをマスタ表形式データと決定し、それ以外の表形式データをスレイブ表形式データと決定するステップと、

e) 前記選択された表形式データにおいて、前記特定された情報ブロックに含まれる値リストを相互に等価にした場合に、マスタ側のレコードに対応するスレイブ側のレコードを対応付けるステップと、

f) 前記二つの表形式データのうちのスレイブ側の表形式データをマスタ側とするような、他の二つの表形式データを選択し、当該他の二つの表形式データについて、前記c) 項目を見出すステップ、d) マスタ表形式データおよびスレイブ表形式データを決定するステップ、および、e) 対応付けるステップとを実行するステップと、

g) 前記f) のステップを、ジョインが必要な表形式データに関して繰り返すステップと、

h) ジョインにより結合された表形式データのうち、根となるものを選択し、前記表形式データの結合にしたがって、各表形式データの深さを決定するステップ

と、

i) 前記深さおよびレコード番号の組み合わせを値とするツリー記述テーブルのための領域を確保するステップと、

j) 初期的には最も浅い深さとなるような、ある深さのレコードを特定し、前記レコードを示す値を、深さとともに、前記ツリー記述テーブルのための領域に配置するステップと、

k) 前記特定されたレコードから、スレイブ側の表形式データのレコードを特定し、当該レコードに基づき、スレイブ側の表形式データがマスタ側となるような表形式データのレコードを示す値を、前記マスタ側の表形式データの深さとともに、前記ツリー記述テーブルのための領域に配置するステップと、

l) 前記 k) のステップを、表形式データが存在しない深さになるまで、或いは、レコードが存在しなくなるまで繰り返すステップと、

m) 前記 j) ないし l) のステップを繰り返すことにより、ツリー記述テーブルのための領域に、所定の深さおよびレコードを特定する値を配置することにより、ツリー記述テーブルを完成させるステップとを備えたことを特徴とする方法。

【請求項 2】 前記 e) のステップにおいて、

前記値リストを等価にする際に、マスタ側の情報ブロックにおいて、その項目値の追加にともなって、前記情報ブロックのポインタ配列を指し示すための第 2 の射影配列を生成するステップと、

前記スレイブ側の情報ブロックにおいて、共通化された項目の値の存在する数を累計化した第 3 の射影配列を生成するステップと、

スレイブ側の情報ブロックにおいて、前記共通化された項目にてソートされたレコードを指し示す値の集合である第 4 の射影配列を生成するステップとを備え、

前記第 2 の射影配列、マスタ側の情報ブロックのポインタ配列、第 3 の射影配列および第 4 の射影配列を順次たどることにより、スレイブ側のレコードが特定されることを特徴とする請求項 1 に記載の方法。

【請求項 3】 さらに、

n) 最も浅い深さの表形式データを除き、マスタ側に関する第 2 の射影配列の逆

写像配列を生成するステップを備え、

前記第4の射影配列からの要素が指し示す逆写像配列の要素により、次の深さの表形式データにおけるマスタ側の第2の射影配列の要素が特定されることを特徴とする請求項2に記載の方法。

【請求項4】 前記逆写像を生成するステップにおいて、

前記マスタ側に関する第2の射影配列の要素数と等しい要素を収容する、逆写像のための配列を配置するステップと、

前記各要素として、レコードを特定し得ない他の値を配置するステップと、

前記第4の射影配列の位置を要素に、かつ、要素を位置とするように、前記逆写像配列に値を付与するステップとを備えたことを特徴とする請求項3に記載の方法。

【請求項5】 さらに、前記項目を共通化すべき二つの表形式データの少なくとも一方において、前記表形式データの共通化すべき項目について、検索、集計およびソートの何れかを施すステップを備えたことを特徴とする請求項1ないし4の何れか一項に記載の方法。

【請求項6】 各々が、項目とこれに含まれる項目値とを含むレコードの配列として表わされる複数の表形式データを結合し、結合された表形式データであるジョインテーブルをツリー構造に変換するプログラムであって、

- a) 各表形式データを、各々が、特定の項目に属する項目値に対応した項目値番号の順に当該項目値が格納されている値リストと、一意的なレコード番号の順に、当該項目値番号を指示するためのポインタ値が格納されたポインタ配列とからなる一以上の情報ブロックに分割するように構成するステップと、
- b) 項目を共通化すべき二つの表形式データを選択するステップと、
- c) 当該選択された二つの表形式データにおいて、共通化すべき項目を見出すステップと、
- d) 前記特定された情報ブロックのうち、デフォルトのソート順が反映される表形式データをマスタ表形式データと決定し、それ以外の表形式データをスレイブ表形式データと決定するステップと、
- e) 前記選択された表形式データにおいて、前記特定された情報ブロックに含ま

れる値リストを相互に等価にした場合に、マスタ側のレコードに対応するスレイブ側のレコードを対応付けるステップと、

f) 前記二つの表形式データのうちのスレイブ側の表形式データをマスタ側とするような、他の二つの表形式データを選択し、当該他の二つの表形式データについて、前記 c) 項目を見出すステップ、d) マスタ表形式データおよびスレイブ表形式データを決定するステップ、および、e) 対応付けるステップとを実行するステップと、

g) 前記 f) のステップを、ジョインが必要な表形式データに関して繰り返すステップと、

h) ジョインにより結合された表形式データのうち、根となるものを選択し、前記表形式データの結合にしたがって、各表形式データの深さを決定するステップと、

i) 前記深さおよびレコード番号の組み合わせを値とするツリー記述テーブルのための領域を確保するステップと、

j) 初期的には最も浅い深さとなるような、ある深さのレコードを特定し、前記レコードを示す値を、深さとともに、前記ツリー記述テーブルのための領域に配置するステップと、

k) 前記特定されたレコードから、スレイブ側の表形式データのレコードを特定し、当該レコードに基づき、スレイブ側の表形式データがマスタ側となるような表形式データのレコードを示す値を、前記マスタ側の表形式データの深さとともに、前記ツリー記述テーブルのための領域に配置するステップと、

l) 前記 k) のステップを、表形式データが存在しない深さになるまで、或いは、レコードが存在しなくなるまで繰り返すステップと、

m) 前記 j) ないし l) のステップを繰り返すことにより、ツリー記述テーブルのための領域に、所定の深さおよびレコードを特定する値を配置することにより、ツリー記述テーブルを完成させるステップとを、前記コンピュータに実行させることを特徴とするプログラム。

【請求項 7】 前記 e) のステップにおいて、

前記値リストを等価にする際に、マスタ側の情報ブロックにおいて、その項目

値の追加にともなう、前記情報ブロックのポインタ配列を指し示すための第2の射影配列を生成するステップと、

前記スレイブ側の情報ブロックにおいて、共通化された項目の値の存在する数を累計化した第3の射影配列を生成するステップと、

スレイブ側の情報ブロックにおいて、前記共通化された項目にてソートされたレコードを指し示す値の集合である第4の射影配列を生成するステップとを、前記コンピュータに実行させ、

前記第2の射影配列、マスタ側の情報ブロックのポインタ配列、第3の射影配列および第4の射影配列を順次たどることにより、スレイブ側のレコードが特定されるよう、前記コンピュータを動作させることを特徴とする請求項6に記載のプログラム。

【請求項8】 さらに、

n) 最も浅い深さの表形式データを除き、マスタ側に関する第2の射影配列の逆写像配列を生成するステップを、前記コンピュータに実行させ、

前記第4の射影配列からの要素が指し示す逆写像配列の要素により、次の深さの表形式データにおけるマスタ側の第2の射影配列の要素が特定されるように、前記コンピュータを動作させることを特徴とする請求項7に記載のプログラム。

【請求項9】 前記逆写像を生成するステップにおいて、

前記マスタ側に関する第2の射影配列の要素数と等しい要素を収容する、逆写像のための配列を配置するステップと、

前記各要素として、レコードを特定し得ない他の値を配置するステップと、

前記第4の射影配列の位置を要素に、かつ、要素を位置とするように、前記逆写像配列に値を付与するステップとを、前記コンピュータに実行させることを特徴とする請求項8に記載のプログラム。

【請求項10】 さらに、前記項目を共通化すべき二つの表形式データの少なくとも一方において、前記表形式データの共通化すべき項目について、検索、集計およびソートの何れかを施すステップを、前記コンピュータに実行させることを特徴とする請求項6ないし9の何れか一項に記載のプログラム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、連鎖したジョインテーブルをツリー構造に、高速変換する手法に関する。

【0002】

【従来の技術】

データベースは種々の用途に用いられているが、中規模ないし大規模システムにおいては、論理的な矛盾が排除できるリレーショナルデータベース（RDB）の使用が主流となっている。RDBは、基本的には、項目および値と、当該値と関連付けられた他の項目の値へのポインタとから構成されているため、これをツリー構造化することは容易ではなかった。特に、大量のデータを取り扱う必要がある場合に、RDBでは大量のリレーションが発生し、データ量が著しく大きくなり、かつ、処理にも著しい時間を要する。

【0003】

これに対して、XMLデータベースなどオブジェクト指向データベースが提案されている。このオブジェクト指向データベースは、データをそもそもツリー構造で保存し、その木（ツリー）の枝を順次たどっていくことが可能となる。

【0004】

【発明が解決しようとする課題】

しかしながら、従来のデータベースにおけるツリー構造においては、以下の挙げるような問題点があった。

(1) 根（ルート）が固定される。

従来のツリー構造では、特定のノードから他の特定のノードへのポインタで構成されているため、ノード間のリンクの繋ぎ代えが不可能である。また、ポインタを1方向にしかたどることができない。このため、ツリーは、論理的には、どの要素を根（ルート）としても良いはずであるにもかかわらず、実際には、根が固定されている。

【0005】

(2) ツリー構造が固定される。

各ノードは、様々な属性を有している。たとえば、職場の住所と、住まいの住所という項目がある場合に、職場の住所に基づくツリー（たとえば、日本—東京都—港区・・・）というツリーと、住まいの住所に基づくツリー（たとえば、日本—神奈川県—横浜市・・・）とは、論理的に別々のツリーを構成する。しかしながら、従来のデータベースにおけるツリー構造においては、これら別々のツリーを構築することが不可能であった。

【0006】

(3) 処理速度が著しく遅く、また、ソートや集計が困難である。

検索はツリーを順次たどる必要があるため著しい処理時間を要する。また、同様にツリーを順次たどって要素を特定する必要があるため、ソートや集計が困難である。

【0007】

その一方、本発明者は、特許文献1に記載したような、表形式データを、各々が、特定の項目に属する項目値に対応した項目値番号の順に当該項目値が格納されている値リストと、一意的なレコード番号の順に、当該項目値番号を指示するためのポインタ値が格納されたポインタ配列とからなる一以上の情報ブロックに分割し、レコード番号、ポインタ配列中、当該レコード番号に対応する位置の要素（ポインタ値）、および、値リスト中、ポインタ値が指し示す位置の要素（項目値）を順次たどることにより、レコード番号に対応する項目値を特定する技術を考案している。

【0008】

【特許文献1】

国際公開第WO00/10103号パンフレット（第3図および第4図）

このようなデータ形式を採用することにより、従来のRDBと比較して、検索、集計、ソートなどを著しく高速に実現することができた。

また、特許文献2に開示したように、上記データ形式を採用した複数のテーブルをジョインする手法も、本発明者は提案している。

【0009】

【特許文献2】

特開 2000-339390 号公報 (第(14)頁～第(17)頁および図 10 ないし図 18)

これにより、項目値が共通する項目をキーとして、複数のテーブルをジョインすることが可能となり、かつ、当該ジョインされたテーブルの検索、集計なども容易に実現することができる。

【0010】

本発明者は、上記特許文献 1 および特許文献 2 を利用して、1 以上のツリー構造を構築することができることに知見し、これを実現した。この方法およびプログラムによれば、上述した (1) ～ (3) の問題点が解決され、任意の根（ルート）から、種々の項目のツリー構造を構築できた。かつ、その処理速度は高速であり、また、集計やソートを実現することもできた。

つまり、本発明は、任意の根を選択して、所望の 1 以上のツリー構造を高速に構築することができ、検索、集計、ソートが高速に実行できる方法およびプログラムを提供することを目的とする。

【0011】

【課題を解決するための手段】

本発明の目的は、各々が、項目とこれに含まれる項目値とを含むレコードの配列として表わされる複数の表形式データを結合し、結合された表形式データであるジョインテーブルをツリー構造に変換する方法であって、

- a) 各表形式データを、各々が、特定の項目に属する項目値に対応した項目値番号の順に当該項目値が格納されている値リストと、一意的なレコード番号の順に、当該項目値番号を指示するためのポインタ値が格納されたポインタ配列とからなる一以上の情報ブロックに分割するように構成するステップと、
- b) 項目を共通化すべき二つの表形式データを選択するステップと、
- c) 当該選択された二つの表形式データにおいて、共通化すべき項目を見出すステップと、
- d) 前記特定された情報ブロックのうち、デフォルトのソート順が反映される表形式データをマスタ表形式データと決定し、それ以外の表形式データをスレイブ表形式データと決定するステップと、

- e) 前記選択された表形式データにおいて、前記特定された情報ブロックに含まれる値リストを相互に等価にした場合に、マスタ側のレコードに対応するスレイブ側のレコードを対応付けるステップと、
- f) 前記二つの表形式データのうちのスレイブ側の表形式データをマスタ側とするような、他の二つの表形式データを選択し、当該他の二つの表形式データについて、前記 c) 項目を見出すステップ、d) マスタ表形式データおよびスレイブ表形式データを決定するステップ、および、e) 対応付けるステップとを実行するステップと、
- g) 前記 f) のステップを、ジョインが必要な表形式データに関して繰り返すステップと、
- h) ジョインにより結合された表形式データのうち、根となるものを選択し、前記表形式データの結合にしたがって、各表形式データの深さを決定するステップと、
- i) 前記深さおよびレコード番号の組み合わせを値とするツリー記述テーブルのための領域を確保するステップと、
- j) 初期的には最も浅い深さとなるような、ある深さのレコードを特定し、前記レコードを示す値を、深さとともに、前記ツリー記述テーブルのための領域に配置するステップと、
- k) 前記特定されたレコードから、スレイブ側の表形式データのレコードを特定し、当該レコードに基づき、スレイブ側の表形式データがマスタ側となるような表形式データのレコードを示す値を、前記マスタ側の表形式データの深さとともに、前記ツリー記述テーブルのための領域に配置するステップと、
- l) 前記 k) のステップを、表形式データが存在しない深さになるまで、或いは、レコードが存在しなくなるまで繰り返すステップと、
- m) 前記 j) ないし l) のステップを繰り返すことにより、ツリー記述テーブルのための領域に、所定の深さおよびレコードを特定する値を配置することにより、ツリー記述テーブルを完成させるステップとを備えたことを特徴とする方法により達成される。

【0012】

本発明によれば、二つの表形式データの組を、連鎖（チェーン）して、深さの浅い表形式データのレコードから、深い位置にある表形式データのレコードを、順次特定していく。これにより、深さとレコードを示す値とからなるツリー記述テーブルを作ることができる。ツリー記述テーブルの深さにより、表形式データが特定され、レコードを示す値により、当該表形式データの実際の値を取得することが可能となる。表形式データの選択や、チェーンのための表形式データの組み合わせ、さらには、ルートとなる表形式データの選択は、自由である。したがって、きわめて柔軟性の高いツリー構造を構築することが可能となる。

【0013】

また、同じ表形式データに関して、複数の異なるツリー記述テーブルを持つことができる。したがって、同じデータについて、複数のツリー構造を構築することも可能となる。

【0014】

好ましい実施態様においては、前記 e) のステップにおいて、

前記値リストを等価にする際に、マスタ側の情報ブロックにおいて、その項目値の追加にともなって、前記情報ブロックのポインタ配列を指し示すための第2の射影配列を生成するステップと、

前記スレイブ側の情報ブロックにおいて、共通化された項目の値の存在する数を累計化した第3の射影配列を生成するステップと、

スレイブ側の情報ブロックにおいて、前記共通化された項目にてソートされたレコードを指し示す値の集合である第4の射影配列を生成するステップとを備え、

前記第2の射影配列、マスタ側の情報ブロックのポインタ配列、第3の射影配列および第4の射影配列を順次たどることにより、スレイブ側のレコードが特定される。

【0015】

より好ましい実施態様においては、さらに、

n) 最も浅い深さの表形式データを除き、マスタ側に関する第2の射影配列の逆写像配列を生成するステップを備え、

前記第4の射影配列からの要素が指し示す逆写像配列の要素により、次の深さの表形式データにおけるマスタ側の第2の射影配列の要素が特定される。

【0016】

さらに、逆写像を生成するステップにおいて、

前記マスタ側に関する第2の射影配列の要素数と等しい要素を収容する、逆写像のための配列を配置するステップと、

前記各要素として、レコードを特定し得ない他の値を配置するステップと、

前記第4の射影配列の位置を要素に、かつ、要素を位置とするように、前記逆写像配列に値を付与するステップとが設けられているのが望ましい。

別の好ましい実施態様においては、さらに、前記項目を共通化すべき二つの表形式データの少なくとも一方において、前記表形式データの共通化すべき項目について、検索、集計およびソートの何れかを施すステップを備えている。

上記実施態様によれば、検索、集計、ソートなどが施された表形式データを使ったツリー構造の構築が可能となる。

【0017】

このように、本発明によれば、任意の根を選択し、かつ、所望のツリー構造を作成することができる。また、ツリー構造を作成するための基となるデータとして、検索、集計、ソートを施したものを利用することも可能となる。

【0018】

また、本発明の目的は、各々が、項目とこれに含まれる項目値とを含むレコードの配列として表わされる複数の表形式データを結合し、結合された表形式データであるジョインテーブルをツリー構造に変換するプログラムであって、

- a) 各表形式データを、各々が、特定の項目に属する項目値に対応した項目値番号の順に当該項目値が格納されている値リストと、一意的なレコード番号の順に、当該項目値番号を指示するためのポインタ値が格納されたポインタ配列とからなる一以上の情報ブロックに分割するように構成するステップと、
- b) 項目を共通化すべき二つの表形式データを選択するステップと、
- c) 当該選択された二つの表形式データにおいて、共通化すべき項目を見出すステップと、

d) 前記特定された情報ブロックのうち、デフォルトのソート順が反映される表形式データをマスタ表形式データと決定し、それ以外の表形式データをスレイブ表形式データと決定するステップと、

e) 前記選択された表形式データにおいて、前記特定された情報ブロックに含まれる値リストを相互に等価にした場合に、マスタ側のレコードに対応するスレイブ側のレコードを対応付けるステップと、

f) 前記二つの表形式データのうちのスレイブ側の表形式データをマスタ側とするような、他の二つの表形式データを選択し、当該他の二つの表形式データについて、前記 c) 項目を見出すステップ、d) マスタ表形式データおよびスレイブ表形式データを決定するステップ、および、e) 対応付けるステップとを実行するステップと、

g) 前記 f) のステップを、ジョインが必要な表形式データに関して繰り返すステップと、

h) ジョインにより結合された表形式データのうち、根となるものを選択し、前記表形式データの結合にしたがって、各表形式データの深さを決定するステップと、

i) 前記深さおよびレコード番号の組み合わせを値とするツリー記述テーブルのための領域を確保するステップと、

j) 初期的には最も浅い深さとなるような、ある深さのレコードを特定し、前記レコードを示す値を、深さとともに、前記ツリー記述テーブルのための領域に配置するステップと、

k) 前記特定されたレコードから、スレイブ側の表形式データのレコードを特定し、当該レコードに基づき、スレイブ側の表形式データがマスタ側となるような表形式データのレコードを示す値を、前記マスタ側の表形式データの深さとともに、前記ツリー記述テーブルのための領域に配置するステップと、

l) 前記 k) のステップを、表形式データが存在しない深さになるまで、或いは、レコードが存在しなくなるまで繰り返すステップと、

m) 前記 j) ないし l) のステップを繰り返すことにより、ツリー記述テーブルのための領域に、所定の深さおよびレコードを特定する値を配置することにより

、ツリー記述テーブルを完成させるステップとを、前記コンピュータに実行させることを特徴とするプログラムによっても達成される。

【0019】

【発明の実施の形態】

以下、添付図面を参照して、本発明の実施の形態につき説明を加える。図1は、本発明の実施の形態にかかる連鎖的な表形式データのジョイン、および、ジョインされたテーブル（ジョインテーブル）をツリー構造に変換するコンピュータシステムのハードウェア構成を示すブロックダイアグラムである。図1に示すように、このコンピュータシステム10は、通常のものと同様の構成であり、プログラムを実行することによりシステム全体および個々の構成部分を制御するCPU12、ワークデータなどを記憶するRAM(Random Access Memory)14、プログラム等を記憶するROM(Read Only Memory)16、ハードディスク等の固定記憶媒体18、CD-ROM19をアクセスするためのCD-ROMドライバ20、CD-ROMドライバ20や外部ネットワーク（図示せず）と接続された外部端子との間に設けられたインタフェース（I/F）22、キーボードやマウスからなる入力装置24、CRT表示装置26を備えている。CPU12、RAM14、ROM16、外部記憶媒体18、I/F22、入力装置24および表示装置26は、バス28を介して相互に接続されている。

【0020】

本実施の形態にかかる、表形式データを連鎖的に結合（ジョイン）するプログラム、結合されたテーブル（ジョインテーブル）をツリー構造に変換するプログラムなどは、CD-ROM19に収容され、CD-ROMドライバ20に読取られても良いし、ROM16に予め記憶されていても良い。また、いったんCD-ROM19から読み出したものを、外部記憶媒体18の所定の領域に記憶しておいても良い。或いは、上記プログラムは、ネットワーク（図示せず）、外部端子およびI/F22を経て外部から供給されるものであっても良い。

【0021】

また、本実施の形態においては、ツリー構造への高速変換を実現するために、後述するように所定のデータ形式の情報ブロックを生成する必要がある。この情

報ブロック生成プログラムも同様に、CD-ROM19に收容され、ROM16に記憶され、或いは、外部記憶媒体18に記憶されても良い。或いは、これらプログラムは、ネットワーク（図示せず）を介して、外部から供給されても良いことはいうまでもない。また、本実施の形態において、情報ブロック生成プログラムにて生成されたデータ（情報ブロック）は、RAM14に記憶され、或いは、外部記憶媒体18の所定の領域に記憶される。

【0022】

本実施の形態においては、図2および図3に示すように、表形式データ（テーブル）を、所定のデータ形式により保持している。たとえば、図2（a）に示すように、テーブル#0である組織テーブルは、実際の項目値である「組織名」が、50音順（英字ではアルファベット順）でソートされて配置された値リスト（以下、場合によって「VL」と称する。）203と、各レコード番号に対応して、当該レコードが指し示す値リスト中の番号が格納された、値リストへのポインタ配列（以下、場合によって「PVL」と称する。）202とを備えた、情報ブロック201により構成される。レコード番号に対応するPVL202中の要素が示す番号の、VL中の要素（項目値）が、当該レコードに対応する項目値となる。なお、PVL202は、テーブルのレコード数だけ要素があることが理解できるであろう。

【0023】

図2（b）は、「部署」という項目の値（項目値）と、「S氏名」という項目の値（項目値）とから構成される社員所属テーブルを示している。社員所属テーブルは、情報ブロック211および情報ブロック221とから構成される。「部署」の情報ブロック211には、（「組織名」のVLと、その要素が共通する「部署名」がソートされて格納されたVL213と、VL213中の要素（部署名）を指し示すためのPVL212とが含まれる。また、「S氏名」の情報ブロック221は、氏名が、ソートされて格納されたVL223と、VL223中の要素（氏名）を指し示すためのPVL222とが含まれる。

【0024】

同様に、図3に示す容易、社員履歴テーブルも、「R氏名」という項目の値か

ら構成される情報ブロック 301 と、「プロジェクト」という項目の値から構成される情報ブロック 311 とから構成される。図 2 (a)、(b) に示す例と同様に、図 3 の例でも、情報ブロックは、それぞれ、PVL (符号 302、312 参照) と、VL (303、313 参照) とを含む。

【0025】

ここで、組織テーブルと社員所属テーブルとを参照すると、組織テーブルの「組織名」と、社員所属テーブルの「部署」とを共有できることがわかる。また、社員所属テーブルと、社員履歴テーブルとを参照すると、社員所属テーブルの「S 氏名」と、社員履歴テーブルの「R 氏名」とを共有できることがわかる。本実施の形態においては、項目を共有させたジョインテーブルを連鎖的に生成することにより、ツリー構造を表す。以下、ツリー構造への変換処理につき、より詳細に説明を加える。

【0026】

図 4 は、本実施の形態にかかるツリー構造への変換処理を示すフローチャートである。図 4 において、ジョイン(Join)テーブルの作成(ステップ 301)、2 番目以降のジョイン(Join)テーブルへの逆写像配列の付加(ステップ 302)、および、ツリー記述テーブルのための領域の確保(ステップ 303)が、処理の準備段階を構成する。まず、これらについて詳細に説明を加える。

なお、図 2 および図 3 に示す例では、組織テーブル、社員所属テーブル、および、社員履歴テーブルの 3 つのテーブルが存在する。ここでは、組織テーブルと社員所属テーブルとの間で、「組織名」(「部署」)をキーとしたジョイン(Join#0)が実行され、テーブルを共有する処理、および、社員所属テーブルと社員履歴テーブルとの間で、「S 氏名」(「R 氏名」)をキーとしたジョイン(Join#1)が実行される。

【0027】

図 5 は、組織テーブルと社員所属テーブルとのジョインを説明する図である。ここでは、組織テーブルの項目値の順序が維持された状態で、組織テーブル(符号 501 参照)と社員所属テーブル(符号 502 参照)とがジョインされている。

。

図4にしたがって、まず、組織テーブルと社員所属テーブルとのジョインテーブルが生成される。図5(a)において、組織テーブルの項目値の順序が維持されているため、これを「マスタ側」と称し、社員所属テーブルの側を「スレイブ側」と称する。ここでは、組織テーブル501と、社員所属テーブル中の「部署」のテーブル502とがジョインされ、これがジョイン(Join#0)テーブルとなる。

【0028】

図5(b)に示すように、マスタ側(符号511参照)に関しては、当該マスタ側の各行(レコード)の重複数を累計化した配列(射影配列「i」とも称する)512およびマスタ側の順序付き集合(射影配列「ii」とも称する)513とが生成される。

【0029】

射影配列「i」は、元の組織テーブル(図2(a))参照の行番号が繰り返される数(累計数)が表される。たとえば、1行目(行番号「0」)の要素は、「0」であり、2行目(行番号「1」)の要素は、「3」であることから、元の組織テーブルにおける行番号「0」に位置する「東京本店」という項目値の重複数は、「 $3 - 0 = 3$ 」であることがわかる。また、射影配列「i」において、2行目(行番号「1」)の要素は「3」であり、3行目(行番号「2」)の要素は「5」であることから、元の組織テーブルにおける行番号「1」に対応する「大阪支店」という項目値の重複数は、「 $5 - 3 = 2$ 」であることがわかる。なお、射影配列「i」の末尾(最下行)には、ジョインされたテーブルの行の総数(ジョインした組織テーブルでは「6」)が配置される。

【0030】

射影配列「ii」は、ジョイン(共通化)した組織テーブルの行番号を指定する、元の組織テーブルの行番号が示されている。つまり、この射影配列「ii」は、共通化により項目が追加された場合に、新たな組織テーブルの行番号を指定するために利用される。射影配列「i」512および射影配列「ii」513を参照することにより、ジョインされた組織テーブルに配置される値を特定することができる。たとえば、ジョインされた組織テーブル(Join#0)に関する行番号「0」な

いし「2」の3行について、順序配列「ii」の行番号「0」の要素は、「0」である。したがって、元の組織テーブルの行番号「0」の要素、つまり、「東京本店」が、ジョインされた組織テーブル中に配置されることがわかる。

【0031】

次に、スレイブ側（符号521）においては、ジョイン(Join#0)でキーとなった項目、つまり、「部署」の値ごとに、ジョインテーブルにおける各行の重複数を累計化した配列（射影配列「iii」とも称する）522、および、スレイブ側のジョインでキーとなった項目でソートされた順序付き集合（射影配列「iv」とも称する）523とが生成される。

【0032】

スレイブ側では、ジョイン(Join#0)でキーとなった項目「部署」でソートされているため、射影配列「iii」は符号522にて示すものとなる。射影配列「iii」により、ソート順にしたがって、「部署」という項目中、項目値「大阪支店」の重複数が「 $2 - 0 = 2$ 」、項目値「東京本店」の重複数が「 $5 - 2 = 3$ 」、項目値「名古屋店」の重複数が「 $6 - 5 = 1$ 」となる。

【0033】

また、射影配列「iv」中の値は、元の社員所属テーブルの行番号を示す。射影テーブル「iii」および射影テーブル「iv」により、ジョインされたテーブルの要素と、元の社員所属テーブルの要素とを対応させることができる。たとえば、「射影配列「iii」の行番号「0」および行番号「1」の要素を参照することにより、ソートされた「部署」という項目の最初の要素である「大阪支店」に関して、2つの要素が存在し、それぞれ、元の社員所属テーブルの行番号「1」および「5」に対応することがわかる。

【0034】

このようにして、組織テーブルと社員所属テーブルとの間で、「組織名」という項目および「部署」という項目をキーとしたジョイン(Join#0)が終了すると、社員所属テーブルと社員履歴テーブルとの間で、「S氏名」という項目および「R氏名」という項目をキーとしたジョイン(Join#1)が実行される。

図6は、社員所属テーブルと社員履歴テーブルとのジョインを説明する図であ

る。先に説明したジョイン(Join#0)と同様に、このジョインにおいても、社員所属テーブルの「S氏名」の値の順序が維持された状態となる。したがって、社員所属テーブルがマスタ側となり、社員履歴テーブルがスレイブ側となる。図6 (b) は、社員所属テーブル中の「S氏名」のテーブル601と、社員履歴テーブル中の「R氏名」のテーブル602とがジョインされ、これがジョイン(Join#1)テーブルとなる。

【0035】

図6 (b) は、このジョイン(Join#1)におけるマスタ側611に関する、各行(レコード)の重複数を累計化した配列(射影配列「i」)612、および、順序付き集合(射影配列「ii」)613を示す。また、図6 (c) は、ジョイン(Join#1)におけるスレイブ側612に関する、ジョイン(Join#1)でキーとなった項目「R氏名」の値ごとに、ジョインテーブルにおける各行の重複数を累計化した配列(射影配列「iii」)622、および、ジョインでキーとなった項目でソートされた順序付き集合(射影配列「iv」)623を示す。これらの生成は、図5に示したジョイン(Join#0)のものと同様である。

【0036】

なお、最初のジョイン(Join#0)のスレイブ側の順序付き集合(射影配列「iv」)523と、2番目のジョイン(Join#1)のマスタ側の順序付き集合(射影配列「ii」)613とについて、双方の配列において、同じ値は、同じレコードを指している。これは、対象とする値(要素)が同一のテーブル(社員所属テーブル)だからである。

【0037】

このように、ジョインテーブルの作成処理(図4のステップ401)により、図7に示すような組織テーブル701と社員所属テーブル703との間に介在するジョインテーブル(組織:所属テーブルJoin#1)702が作られ、社員所属テーブル703と社員履歴テーブル705との間に介在するジョインテーブル(所属:履歴テーブルJoin#1)704が作られる。さらに、社員履歴テーブルと、さらに、ジョインすべきテーブルが存在すれば、これらの間に介在する他のジョインテーブル(Join#2)が作られることは言うまでもない。

【0038】

ジョインテーブルの作成が完了すると、最初のジョインテーブル以外のジョインテーブル、つまり、2番目以降のジョインテーブルについて、逆写像配列を生成して、これを付加する（ステップ402）。逆写像配列は、配列中の値により、もとの配列を示す番号を指定するものである。

図8に示すように、逆写像配列を作成するためには、逆写像の元になる配列（写像配列「ii」）と同じレコード数の配列の領域800を確保し（図8（a）参照）、当該配列中の要素として、レコードを示す番号となり得ない値（たとえば、「-1」）を与える（図8（b）参照）。次いで、元の配列（射影配列「ii」613）の値が指し示す番号の要素に、元の配列の番号（レコード番号）を配置する（図8（c）参照）。このようにして、図8（c）の右端にあるような、最終的な写像配列800を得ることが可能となる。

【0039】

次いで、ツリー記述テーブルのための領域が確保される（ステップ403）。図10は、ツリー記述テーブルの例を示す図である。ツリー記述テーブルは、複数の、深さ（Depth）とレコード番号（Rec No）との組み合わせから構成される。初期的には、それぞれの項目には、値が付与されず、以下に説明する処理により、順次、値が付与されていく。深さとは、ツリーノードの深さを表す。図7に示す例においては、「組織テーブル」、「社員所属テーブル」、「社員履歴テーブル」の深さは、それぞれ、「0（ゼロ）」、「1」、「2」である。

【0040】

本実施の形態においては、ツリー構造を記述したツリー記述テーブルを作成するために、図9に示すように、確保されたツリー記述テーブルのための領域と、第1のジョイン（Join#0）、つまり、組織テーブルと社員所属テーブルとの間のジョインに関する第1の配列群910と、第2のジョイン（Join#1）、つまり、社員所属テーブルと社員履歴テーブルとの間のジョインに関する第2の配列群920と、これらの間を結びつける逆写像配列921とが利用される。

【0041】

第1の配列群910は、第1のジョイン（Join#0）の順序付集合（射影配列「ii

」) 911、ジョインキーのPVL912、累計数配列(射影配列「iii」) 913および順序付集合(射影配列「iv」) 914を含む。また、第2の配列群920も、同様に、第2のジョイン(Join#1)の順序付集合(射影配列「ii」) 922、ジョインキーのPVL923、累計数配列(射影配列「iii」) 924および順序付集合(射影配列「iv」) 925を含む。

【0042】

まず、深さが「0(ゼロ)」に初期化された後(ステップ404)、現在の深さの要素が取得される(ステップ405)。要素が取得された場合には(ステップ406でイエス(Yes))、現在の深さの要素が、その深さとともにツリー記述テーブルに登録される(ステップ407)。上記例では、「組織テーブル」中の要素が取得される。図11において、順序付集合(射影配列「ii」)の先頭(行番号「0」)の要素が、その深さ「0」とともに、ツリー記述テーブルの最初の要素として配置される(符号1000参照)。

【0043】

次いで、「子」がいるか否かが判断される(ステップ408)。ここに、「子」とは、ある深さのテーブル(たとえば、深さ「0」のテーブル(組織テーブル))の順序付集合(射影配列「ii」)から配列をたどることで達する、次の深さのテーブル(たとえば、深さ「1」のテーブル)中の順序付集合(射影配列「ii」)中の対応する要素(つまり、次の深さのテーブルのレコード番号)をいう。図11の例では、図12に示すように、順序付集合(射影配列「ii」)の行番号「0」の要素「0」は、深さ「0」のテーブル(組織テーブル)のPVL、累計数配列(射影配列「iii」)および順序付集合(射影配列「iv」)、並びに、逆写像配列および深さ「1」のテーブル(社員所属テーブル)の順序付集合(射影配列「ii」)を順にたどっていくことで、要素が存在することがわかる。したがって、ステップ408でイエス(Yes)と判断される。

【0044】

この場合には、深さがインクリメントされ(ステップ409)、当該深さ(上記例では「1」)の要素が取得され、深さ(上記例では「1」)とともに、取得された要素が、ツリー記述テーブルの次の要素として配置される(符号1001

参照)。なお、最初のジョイン(Join#0)における射影配列「iv」の要素、および、これに引き続いているジョイン(Join#1)における射影配列「ii」の要素は、双方とも、同じテーブルのレコードを指している。したがって、これらが同一であれば、同じレコードを指していることに留意されたい。

【0045】

図12において、深さ「0」のテーブル（組織テーブル）の累計数配列（射影配列「iii」）において、行番号「1」の要素「2」と、次の行番号「2」の要素「5」とから、3個の要素が存在することがわかる（符合1201参照）。つまり、この深さ「0」において、上記累計数配列の行番号「1」の要素は、次に位置する順序付集合（射影配列「iv」）の行番号「2」、「3」、「4」の3つを指し示していることがわかる。なお、図12以降、累計数配列（射影配列「iii」）の右側に付与された矢印は、矢印の先に位置する要素と、矢印の元に位置する要素との差だけ、矢印の元に位置する要素が指し示すことができる、順序付集合（射影配列「iv」）の要素が存在することを意味している。

【0046】

次いで、さらに深さがインクリメントされ（ステップ409）、次の深さの要素が取得される（ステップ405）。図12の例に関しては、深さがインクリメントされて「2」となり、深さ「1」のテーブルの順序付集合（射影配列「ii」）、PVL、累計数配列（射影配列「iii」）および順序付集合（射影配列「iv」）をたどる。図13に示すように、順序付集合（射影配列「ii」）の行番号「0」の要素をたどると、最終的に、順序付集合（射影集合「iv」）の行番号「0」、「1」のそれぞれの要素「0」、「7」が取り出される。これは、累計数配列（射影配列「iii」）の関連する要素が「0」で、かつ、次の行番号の要素が「2」であるため、要素が「 $2 - 0 = 2$ 」だけあることがわかるからである。したがって、これら順序付集合（射影配列「iv」）の行番号「0」および「1」の要素、「0」および「7」が、それぞれ、深さ「2」とともに、ツリー記述テーブルの要素として配置される（符号1002、1003参照）。

【0047】

これ以上の深さ（深さ「3」）は存在しないため、深さ方向の探索が終了する

。つまり、ステップ405～ステップ409のループにおいて、次のステップ406においてノー(No)と判断され、深さがデクリメントされる(ステップ410)。つまり、ひとつ浅い深さに戻り、当該深さが「0」以上であれば処理が続行される(ステップ411参照)。

【0048】

上記例では、深さが「2」から「1」になるため、当該深さ「1」の要素が再度取得される。図12の符号1201に関連して説明したように、深さ「1」の累計数配列(射影配列「iii」)に関して、3つの要素が存在することがわかっている。これらのうち、2つ目の要素および3つ目の要素については未処理である。したがって、次のステップ405においては、上記2つめの要素に基づいて処理が実行される。図14に示すように、順序付集合(射影配列「iv」)の2つ目の要素(行番号「2」の次に位置する行番号「3」の要素)から、逆写像配列、および、次の順序付集合(射影配列「ii」)がたどられ、対応する要素が取得される。これが、深さ「1」とともに、ツリー記述テーブルの要素として配置される(符号1004参照)。

【0049】

さらに、深さがインクリメントされて(ステップ409参照)、深さ「2」の要素を取得する処理が実行される。図15に示すように、順序付集合(射影配列「ii」)、PVL、累計数配列(射影配列「iii」)および順序付集合(射影配列「iv」)をたどることで、深さ「2」の要素が取得される。なお、累計数配列に配置された矢印から理解できるように、ここでは、2つの要素が取得される。取得された要素は、それぞれ、深さ「2」とともに、ツリー記述テーブルの要素として配置される(符号1005、1006)。なお、深さ「3」は存在しないため、深さ方向の探索は終了する。

【0050】

同様に、図16に示すように、深さ「1」の3つ目の要素(順序付集合(射影配列「iv」)の行番号「4」の要素)から、逆写像配列および順序付集合(射影配列「ii」)をたどることで、深さ「1」の要素を取得することができる。これも、その深さ「1」とともにツリー記述テーブルに配置される(符号1007参照)

）。さらに、図17に示すように、深さ「2」に関しても、配列をたどることにより、要素が取得され、その深さ「2」とともに、ツリー記述テーブルに配置される（符号1008参照）。なお、ここでも、深さ「3」は存在しないため、深さ方向の探索は終了する。

【0051】

このような終了の後、深さをデクリメントして「1」としても要素が取得できないため（ステップ410、411、415および406）、さらに深さがデクリメントされて「0」とされて、深さ「0」の要素が取得される。図18に示すように、順序付集合（射影配列「ii」）の次（つまり、行番号「1」）の要素「1」が、ツリー記述テーブルに登録すべき要素となる。したがって、深さ「0」とともに、これがツリー記述テーブルに配置される（符号1009参照）。その後、図4、および、図11ないし図18を参照して説明したものと同様に、深さ方向への順次探索、および、要素のツリー記述テーブルへの配置が繰り返される。

【0052】

図19は、このような処理を続けた結果、最終的なツリー記述テーブルの要素が取得された状態を示す図である。ここに示す処理は、図示しない先の処理にて、深さ「1」の要素をツリー記述テーブルに配置したことに引き続いて実行される。ツリー記述テーブルに配置された要素は、当該深さ「1」に関する「Join#1」側の順序付集合（射影配列「ii」）と共通の意味を持つ。したがって、この要素をたどることで、深さ「2」の要素を探索することができる。

図19の例では、順序付集合（射影配列「ii」）の行番号「2」の要素「2」から、PVLおよび累計数配列（射影配列「iii」）をたどり、順序付集合（射影配列「iv」）の行番号「4」の要素である「4」が取得され、これが、深さ「2」とともに、ツリー記述テーブルに配置される。

【0053】

このようにして完成したツリー記述テーブル900の意味につき、図20を参照して説明する。前述したように、ツリー記述テーブル900には、深さ（Depth）とレコード番号（Rec No）との組み合わせが配置されている。たとえば、深

さの項目の値は、それがどの深さのテーブルに関連するかを特定し、かつ、レコード番号は、深さにて特定されたテーブルのレコード番号を特定している。

【0054】

たとえば、ツリー記述テーブルの最初の要素の組み合わせ(0, 0)により、深さ「0」のテーブル(組織テーブル)の、レコード番号「0」の要素、つまり、「東京本店」が示される。また、これに引き続く要素の組み合わせ(1, 0)により、深さ「1」のテーブル(社員所属テーブル)のレコード番号「0」の要素、つまり、「東京本店」および「A氏」を示している。同様に、さらに引き続く要素の組み合わせ(2, 0)、(2, 7)は、それぞれ、深さ「2」のテーブル(社員履歴テーブル)の、それぞれ、レコード番号「0」および「7」の要素、つまり、「A氏」および「Prj-1」、並びに、「A氏」および「Prj-4」を示すことになる。このように、ツリー記述テーブルは、複数のジョインされたテーブルをツリー構造にて記述したことが理解できるであろう。

このように、本実施の形態によれば、所望のテーブルを指定することで、上述した処理により、ツリー記述テーブルを作成し、ジョインされたテーブルをツリー構造にて記述することが可能となる。

【0055】

次に、本発明の第2の実施の形態につき説明を加える。第1の実施の形態においては、表形式データの全体集合をジョインしたもののツリー記述テーブルを生成したが、第2の実施の形態では、一部の表形式データを、所定の項目で検索し、かつ、ソートした部分集合を用いている。なお、第2の実施の形態においても、システムのハード構成(図1)およびツリー構造への変換処理の概略(図4)については、第1の実施の形態のものとはほぼ同様である。

【0056】

図21～図22は、第2の実施の形態にかかる表形式データ(テーブル)の例を示す図である。第2の実施の形態においては、組織テーブルおよび社員履歴テーブルは、第1の実施の形態と同様のものを利用し(図21の符号201および図23の符号301, 311参照)、かつ、ジョインの対象は全体集合としている(図21の符号2101および図23の符号2301参照)。その一方、社員

所属テーブル自体は、第1の実施の形態のものと同様であるが（図22の符号211、221参照）が、テーブルを「部署」の項目の「大阪支店」、「名古屋支店」でOR検索し（図22の符号2202参照）、かつ、項目「S氏名」でソートしている（図22の符号2203参照）。

【0057】

これら表形式データの間で、「組織名」（「部署」）をキーとしたジョイン（Join#0）は、全体集合（符号2101、2201参照）の間のジョインであり、その一方、「S氏名」（「R氏名」）をキーとしたジョイン（Join#1）は、検索およびソートされた部分集合（図22の符号2203参照）と全体集合（図23の符号2301参照）との間のジョインとなる。

【0058】

「組織名」（「部署」）をキーとしたジョイン（Join#0）に関して作成される配列は、全体集合に関するものであるため、図5に示すものと同様になる。その一方、「S氏名」（「R氏名」）をキーとしたジョイン（Join#1）は、図24に示すようなものとなる。図24に示すように、このジョイン（Join#1）では、項目「部署」の「大阪支店」、「名古屋店」でOR検索され、項目「S氏名」でソートされた社員所属テーブル（マスタ側）2401、および、上記OR検索により抽出された「S氏名」と同一の「R氏名」をもつような社員履歴テーブル（スレイブ側）2402に関連して種々の配列が作成される。

【0059】

図24（b）に示すように、マスタ側に関しては、マスタ側の各行の重複数を類型化した射影配列「i」2412、および、マスタ側の順序付き集合である射影配列「ii」2413が生成される。その一方、図24（c）に示すように、スレイブ側に関しては、ジョイン（Join#1）でキーとなった項目、つまり、「R氏名」の値ごとに、ジョインテーブルにおける各行の重複数を累計化した配列である射影配列「iii」2422、および、上記キーとなった項目でソートされた順序付き集合である射影配列「iv」2423が生成される。これら各種配列およびその要素が示す意味については、図5および図6を参照して説明した、第1の実施の形態のものと同様である。

【0060】

次いで、逆写像配列の作成（図4のステップ402）につき説明を加える。逆写像配列の作成も、基本的には第1の実施の形態と同様である。ただし、逆写像配列の要素数は、社員所属テーブル（マスタ側）の全レコード数に対応する（図25（a）参照）。次いで、図25（b）に示すように、第1の実施の形態と同様に、各要素に、レコード番号となり得ない値（たとえば、「-1」）を付与し、その後、図25（c）に示すように、元の配列の値が指し示す番号の要素に、元の配列の番号（レコード番号）を配置する。これにより、図25（c）の右端に示すような逆写像配列2500を得ることができる。第2の実施の形態においては、逆写像配列の要素中、レコード番号を示していないものが含まれることに留意すべきである。

【0061】

このようにして、種々の射影配列および逆写像配列が生成された後に、図4のステップ404～411の処理が繰り返され、ツリー記述テーブルが出来上がる。図26は、ツリー記述テーブルを作成するために利用される配列群などを示す図である。図26に示すように、ツリー記述テーブルを作成するために、確保されたツリー記述テーブルのための領域と、第1のジョイン(Join#0)、つまり、組織テーブルと社員所属テーブルとの間のジョインに関する第1の配列群2610と、第2のジョイン(Join#1)、つまり、社員所属テーブルと社員履歴テーブルとの間のジョインに関する第2の配列群2620と、これらの間を結びつける逆写像配列2621とが利用される。

【0062】

第1の配列群2610は、図9に示す第1の配列群910と同様であり、第1のジョイン(Join#0)の順序付集合（射影配列「ii」）2611、ジョインキーのPVL2612、累計数配列（射影配列「iii」）2613および順序付集合（射影配列「iv」）2614を含む。また、第2の配列群2620も、同様に、第2のジョイン(Join#1)の順序付集合（射影配列「ii」）2622、ジョインキーのPVL2623、累計数配列（射影配列「iii」）2624および順序付集合（射影配列「iv」）2625を含む。

【0063】

図27に示すように、ツリー記述テーブル2600においては、深さ(Depth)およびレコード番号(Rec No)の組み合わせが、順次配置されていく。

ツリー記述テーブルに要素を配置する処理においては、まず、深さが「0(ゼロ)」に初期化された後(図4のステップ404)、現在の深さの要素が取得され、これが、深さとともに配置される(ステップ405~407)。図28に示すように、初期的には、第1のジョイン(Join#0)の順序付集合(射影配列「ii」)の行番号「0」の要素が、ツリー記述テーブル2600の最初の要素として配置される(符号2700参照)。

【0064】

次いで、図4にしたがって、「子」がいるか否かが判断される(ステップ408)。図29に示すように、第1のジョイン(Join#0)に関して、順序付集合(射影配列「ii」)の行番号「0」の要素から、PVL、累計数配列(射影配列「iii」)および順序付集合(射影配列「iv」)の対応する要素をたどった結果、指し示される逆写像配列の3つの要素がすべて「-1」となる。前述したように逆写像配列の要素「-1」は、レコードが存在しないことを示しているため、この場合には、第2のジョイン(Join#1)に関して、「子」となる要素が存在していないことになる(符号2901参照)。

【0065】

したがって、図4の処理においては、ステップ408でノー(No)と判断され、同じ深さ「0」の次の要素が取得される(ステップ405参照)。上記例では、図30に示すように、順序付集合(射影配列「ii」)の行番号「1」の要素「1」が、その深さ「0」とともにツリー記述テーブルに配置される(符号2701参照)。

【0066】

次いで、「子」となる要素の有無が判断される。上記順序付集合(射影配列「ii」)の要素から、PVL、累計数配列(射影配列「iii」)、順序付集合(射影配列「iv」)、および、逆写像配列の対応する要素をたどった結果、第2のジョイン(Join#1)の順序付集合(射影配列「ii」)の要素が特定される。したがっ

て、この順序付集合（射影配列「ii」）の行番号「1」の要素「1」が、その深さ「1」とともに、ツリー記述テーブルに配置される（符号2702参照）。

【0067】

さらに、「子」となる要素の有無が判断される。上記例では、第2のジョイン（Join#1）の順序付集合の要素から、PVL、累計数配列（射影配列「iii」）、および、順序付集合（射影配列「iv」）の要素が特定され、順序付集合（射影配列「iv」）の2つの要素が、それぞれ、その深さ「2」とともに、ツリー記述テーブルに配置される（符号2703、2704参照）。

【0068】

このような処理を繰り返すことにより、図27に示すようなツリー記述テーブルを完成させることができる。このようにして完成されたツリー記述テーブルの意味も、図20を参照して説明した第1の実施の形態のものと同様である。図33に示すように、ツリー記述テーブルの深さの項目の値は、どの深さのテーブルに関連するかを特定し、かつ、対応するレコード番号は、深さにて特定されたテーブルのレコード番号を特定している。

【0069】

図33において、たとえば、最初の要素の組み合わせ（0，0）により、深さ「0」のテーブル（組織テーブル）の、レコード番号「0」の要素、つまり、「東京本店」が示される。また、これに引き続く、要素の組み合わせ（0，1）により、深さ「0」のテーブル（組織テーブル）の、レコード番号「1」の要素、つまり、「大阪支店」が示される。同様に、さらに引き続く要素の組み合わせ（1，1）および（2，1）により、深さ「1」のテーブル（社員所属テーブル）および深さ「2」のテーブル（社員履歴テーブル）の、それぞれ、レコード番号「1」および「1」の要素、つまり、「大阪支店」および「D氏」、並びに、「D氏」および「Prj-1」が示される。

【0070】

これら実施の形態によれば、任意のノードをルートに設定して、ツリー構造を構築することが可能となる。たとえば、第2の実施の形態においては、図34の符号3400に示すようなツリー構造を構築することができた。本実施の形態を

利用することで、任意の項目をルートとしてツリー構造を再構築することができる。たとえば、図34のうち、社員所属テーブルの行番号「1」のレコードである「大阪支店」および「D氏」をルートと選択して、新たなツリー構造3410を構築することも可能である。

【0071】

ツリー構造を構築する場合には、親ノード群となるレコード群（表形式データ）と、子ノード群となるレコード群（表形式データ）とを選択し、これらをジョインさせればよい。これを所望の回数だけ連鎖的に繰り返した後、ツリー記述テーブルを作成することにより、任意の深さのツリー構造を構築することが可能となる。

【0072】

第1および第2の実施の形態によれば、非常に高速に、複数の表形式データをジョインさせることで、ツリー構造を構築することができる。たとえば、「Pen t i u m（登録商標）4」プロセッサ（1.8GHz）を利用すると、ジョインに関する種々の配列の作成を除き、1000万行／秒ほどの速度で、ツリー記述テーブルに、値を配置することが可能となる。これは、従来の100～1000倍程度の速さである。

【0073】

また、図34を参照して説明したように、任意のノードをルートとしたツリー構造を構築することができ、非常に高い柔軟性を備えている。また、ツリー記述テーブルによりツリー構造が表されるため、同じ表形式データによる複数の異なるツリー構造を保持することが可能となる。たとえば、社員名簿を、所属部署によるツリーで表現するとともに、出生地域によるツリーで表現することが可能となる。

【0074】

さらに、ソートや集計を組み合わせることができる（第2の実施の形態参照）。

また、従来のツリー構造と比較して、使用するメモリ領域をコンパクトにすることが可能となる。たとえば、XMLデータベースにおいては、ノードの接続情

報にパス名を表す文字列を埋め込むため、著しく多量なメモリ領域を必要とする。その一方、本実施の形態によれば、ツリー記述テーブルによりツリー構造を記述することができる。本発明者の実験によれば、株価情報に関して、XMLデータベースと比較して、1/5のメモリ量にてツリー構造を表現できた。

【0075】

本発明は、以上の実施の形態に限定されることなく、特許請求の範囲に記載された発明の範囲内で、種々の変更が可能であり、それらも本発明の範囲内に包含されるものであることは言うまでもない。

たとえば、前記実施の形態においては、3つの表形式データを、連鎖的にジョインしているが、連鎖の数はこれに限定されるものでないことは言うまでもない。

また、第1の実施の形態において、あるジョインと引き続くジョインとの間に、逆写像配列を介在させている。しかしながら、表形式データの全体集合をそのまま利用する場合には、この逆写像配列を省略しても良い。

【0076】

さらに、前記実施の形態においては、一般のコンピュータシステム10内に、所定のプログラムを読み込み、当該プログラムを実行することにより、複数の表形式データのジョインおよびジョインされた表形式データに関する処理を実現しているが、本発明はこれに限定されるものではなく、パーソナルコンピュータ等のような一般のコンピュータシステムに、データベース処理専用のボードコンピュータを接続し、当該ボードコンピュータが上記処理を実行できるように構成しても良いことは言うまでもない。したがって、本明細書において、手段とは必ずしも物理的手段を意味するものではなく、各手段の機能が、ソフトウェアおよび/またはハードウェアによって実現される場合を包含する。さらに、一つの手段の機能が、二つ以上の物理的手段により実現されても、若しくは、二つ以上の手段の機能が、一つの物理的手段により実現されてもよい。

【0077】

本発明は、たとえば、RDBにて保持される部品リストに基づき、部品表(BOM)を作成するための処理(BOM展開)などに利用することができる。また

、本発明は、分流や合流を特徴とするフローにおいて、これを体系的に手際よく管理することが可能となる。たとえば、

ある食品（たとえば、畜肉）が、一頭ごとにどの部位がどのような経路を経て、そこで販売されたかを管理する場合に、これはツリー状のデータとなる。一般に、第1次産業や第2次産業において作られた生産物は、消費されるか、或いは、形を変えるまで、ツリー状に拡散をしていく。本発明は、このように、分流する状態を示すツリー構造を構築し、これを管理することが可能となる。また、本発明によれば、時間の経過に伴いツリーが伸張する状態を、ツリー記述テーブルを作成することにより把握することが可能となる。

【0078】

その一方、ある工業製品を生産するために、様々な部品、エネルギーおよびサービスを購入する場合にも、ツリー状のデータとなる。たとえば、自動車（最終製品）を生産するために、ある部品会社から点火プラグを購入し、部品会社は、電力会社から電力を購入するとともに、点火プラグの部品であるセラミックを、他の部品会社から購入する。このように、時間を遡るのにしたがって、ツリーが拡散する。本発明では、このような合流する状態を示すツリー構造を構築し、これを管理することも可能となる。

【0079】

【発明の効果】

本発明によれば、任意の根を選択して、所望の1以上のツリー構造を高速に構築することができ、検索、集計、ソートが高速に実行できる方法およびプログラムを提供することが可能となる。

【図面の簡単な説明】

【図1】 図1は、本発明の実施の形態にかかるツリー構造の構築方法を実現できるコンピュータシステムのハードウェア構成を示すブロックダイアグラムである。

【図2】 図2は、表形式データの例、および、当該表形式データに基づく情報ブロックの例を示す図である。

【図3】 図3は、表形式データの例、および、当該表形式データに基づく情

報ブロックの例を示す図である。

【図 4】 図 4 は、本実施の形態にかかるツリー構造への変換処理を示すフローチャートである。

【図 5】 図 5 は、組織テーブルと社員所属テーブルとのジョインを説明する図である。

【図 6】 図 6 は、本実施の形態にかかる社員所属テーブルと社員履歴テーブルとのジョインを説明する図である。

【図 7】 図 7 は、本実施の形態にかかるジョインの連鎖（チェーン）の例を示す図である。

【図 8】 図 8 は、本実施の形態にかかる逆写像配列の生成を説明する図である。

【図 9】 図 9 は、本実施の形態にかかるツリー記述テーブルの生成に利用する種々の配列を示す図である。

【図 10】 図 10 は、本実施の形態にかかるツリー記述テーブルの例を示す図である。

【図 11】 図 11 は、第 1 の実施の形態にかかるツリー記述テーブルに値を配置する処理を説明するための図である。

【図 12】 図 12 は、第 1 の実施の形態にかかるツリー記述テーブルに値を配置する処理を説明するための図である。

【図 13】 図 13 は、第 1 の実施の形態にかかるツリー記述テーブルに値を配置する処理を説明するための図である。

【図 14】 図 14 は、第 1 の実施の形態にかかるツリー記述テーブルに値を配置する処理を説明するための図である。

【図 15】 図 15 は、第 1 の実施の形態にかかるツリー記述テーブルに値を配置する処理を説明するための図である。

【図 16】 図 16 は、第 1 の実施の形態にかかるツリー記述テーブルに値を配置する処理を説明するための図である。

【図 17】 図 17 は、第 1 の実施の形態にかかるツリー記述テーブルに値を配置する処理を説明するための図である。

【図 18】 図 18 は、第 1 の実施の形態にかかるツリー記述テーブルに値を配置する処理を説明するための図である。

【図 19】 図 19 は、第 1 の実施の形態にかかるツリー記述テーブルに値を配置する処理を説明するための図である。

【図 20】 図 20 は、第 1 の実施の形態にかかるツリー記述テーブルにより構築されたツリー構造を説明するための図である。

【図 21】 図 21 は、表形式データの他の例、および、当該表形式データに基づく情報ブロックの例を示す図である。

【図 22】 図 22 は、表形式データの他の例、および、当該表形式データに基づく情報ブロックの例を示す図である。

【図 23】 図 23 は、表形式データの他の例、および、当該表形式データに基づく情報ブロックの例を示す図である。

【図 24】 図 24 は、第 2 の実施の形態にかかる社員所属テーブルと社員履歴テーブルとのジョインを説明する図である。

【図 25】 図 25 は、第 2 の実施の形態にかかる逆写像配列の生成を説明する図である。

【図 26】 図 26 は、第 2 の実施の形態にかかるツリー記述テーブルの生成に利用する種々の配列を示す図である。

【図 27】 図 27 は、第 2 の実施の形態にかかるツリー記述テーブルを示す図である

【図 28】 図 28 は、第 2 の実施の形態にかかるツリー記述テーブルに値を配置する処理を説明するための図である。

【図 29】 図 29 は、第 2 の実施の形態にかかるツリー記述テーブルに値を配置する処理を説明するための図である。

【図 30】 図 30 は、第 2 の実施の形態にかかるツリー記述テーブルに値を配置する処理を説明するための図である。

【図 31】 図 31 は、第 2 の実施の形態にかかるツリー記述テーブルに値を配置する処理を説明するための図である。

【図 32】 図 32 は、第 2 の実施の形態にかかるツリー記述テーブルに値を

配置する処理を説明するための図である。

【図 3 3】 図 3 3 は、第 2 の実施の形態にかかるツリー記述テーブルにより構築されたツリー構造を説明するための図である。

【図 3 4】 図 3 4 は、本発明にかかるツリー構造のさらに他の例を示す図である。

【符号の説明】

10	コンピュータシステム
12	CPU
14	RAM
16	ROM
18	固定記憶装置
20	CD-ROMドライバ
22	I/F
24	入力装置
26	表示装置

【書類名】 図面

【図 1】

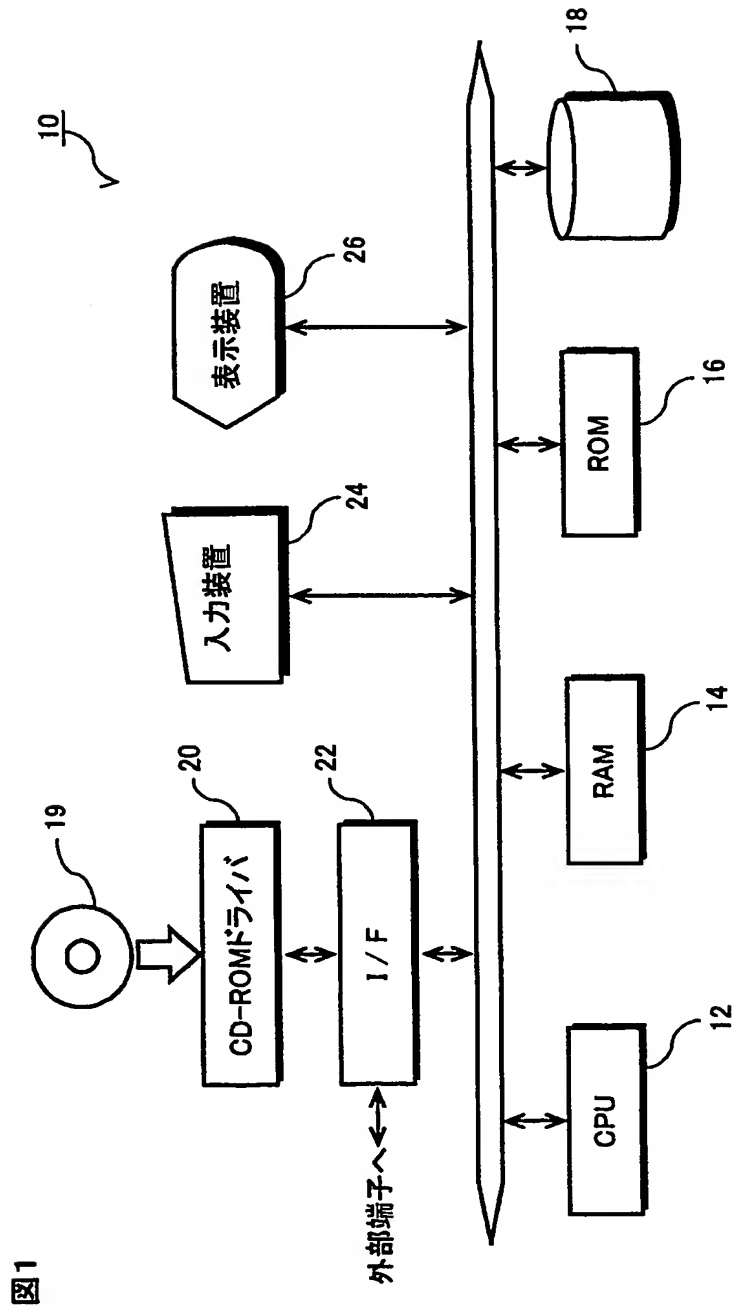


図2

(a)

組織テーブル

	組織名
0	東京本店
1	大阪支店
2	名古屋店

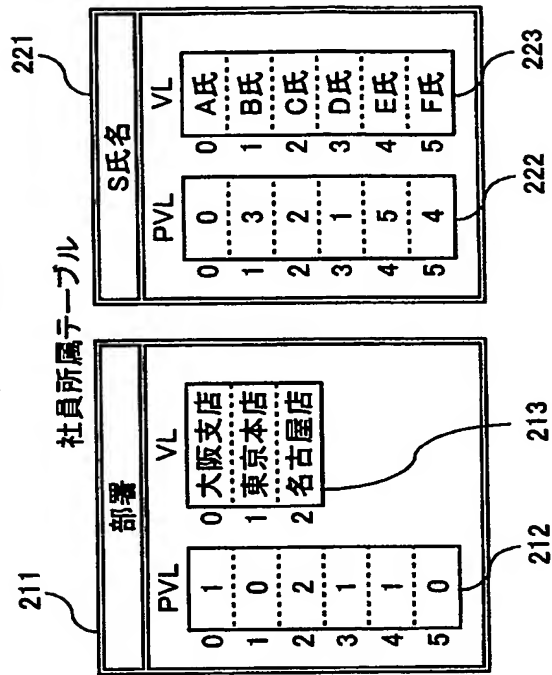
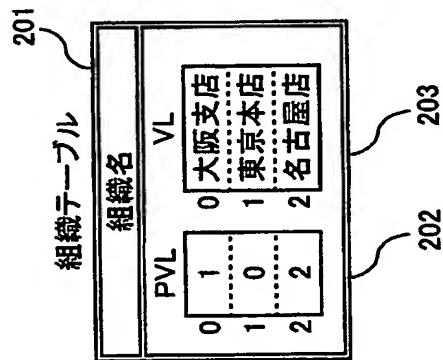
(テーブル#0)

(b)

社員所属テーブル

	部署	S氏名
0	東京本店	A氏
1	大阪支店	D氏
2	名古屋店	C氏
3	東京本店	B氏
4	東京本店	F氏
5	大阪支店	E氏

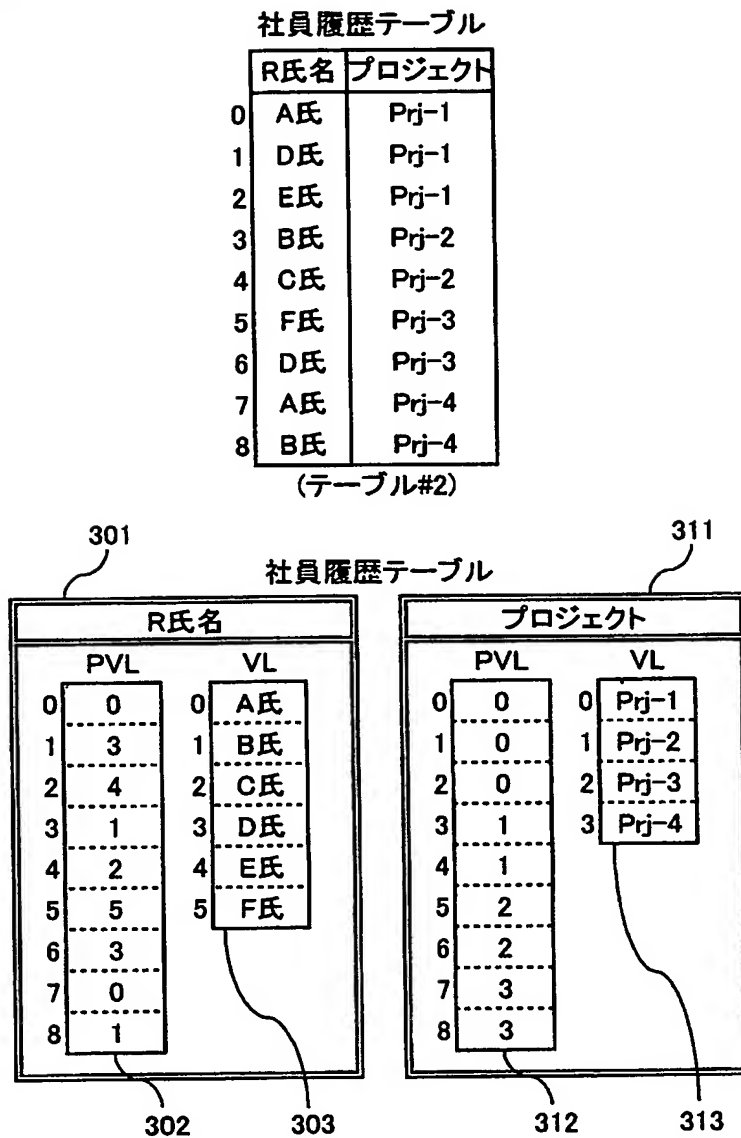
(テーブル#1)



【図2】

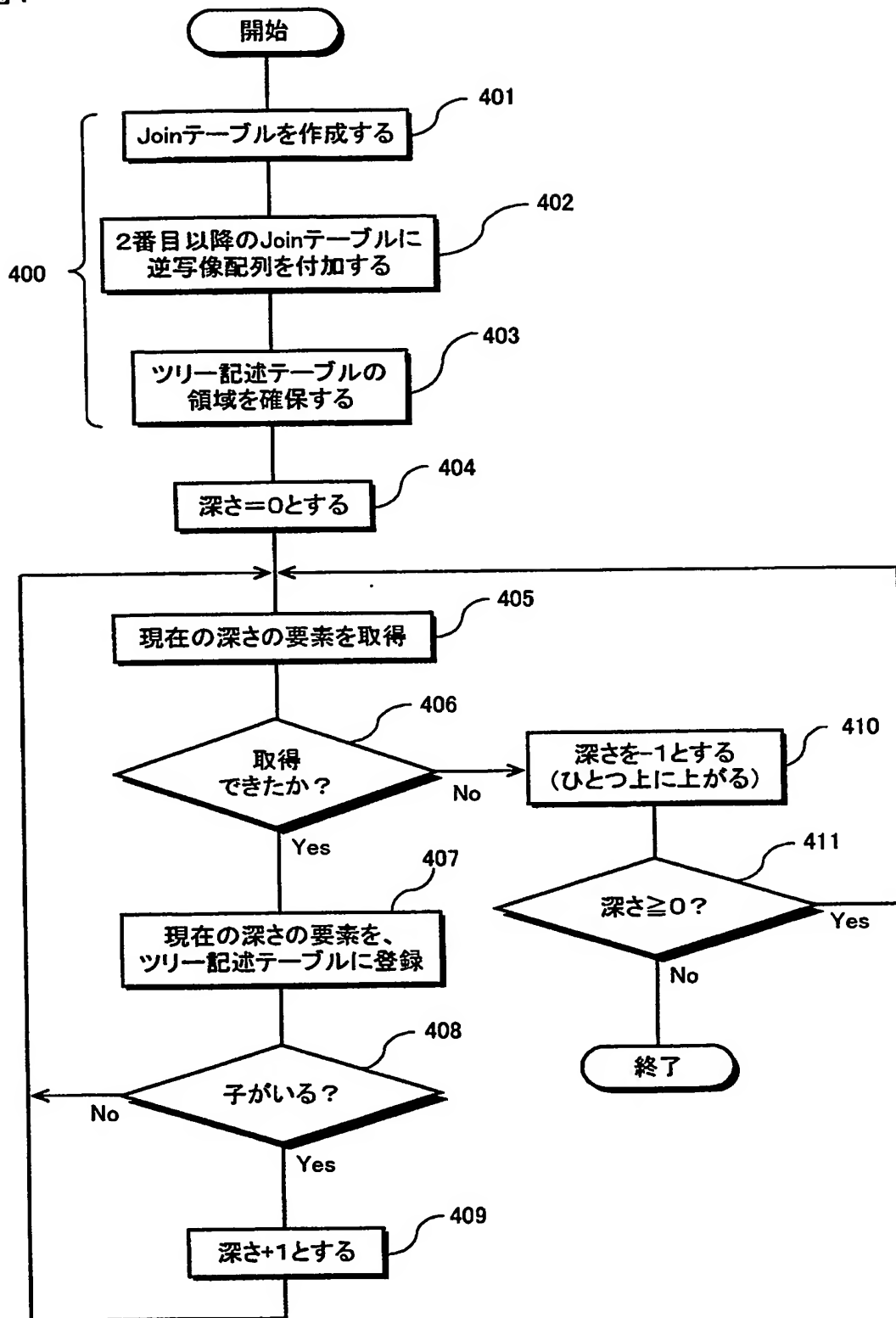
【図 3】

図3

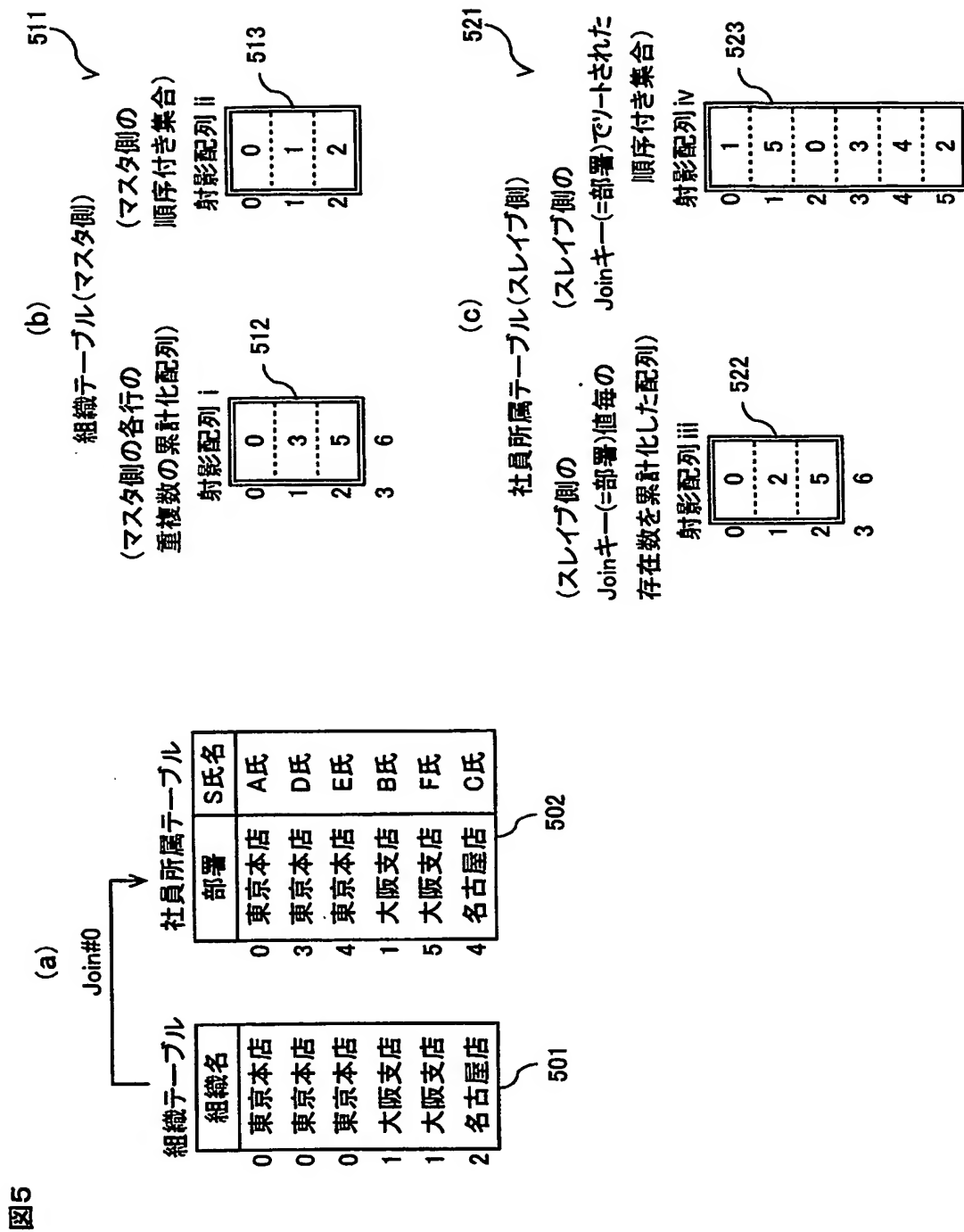


【図 4】

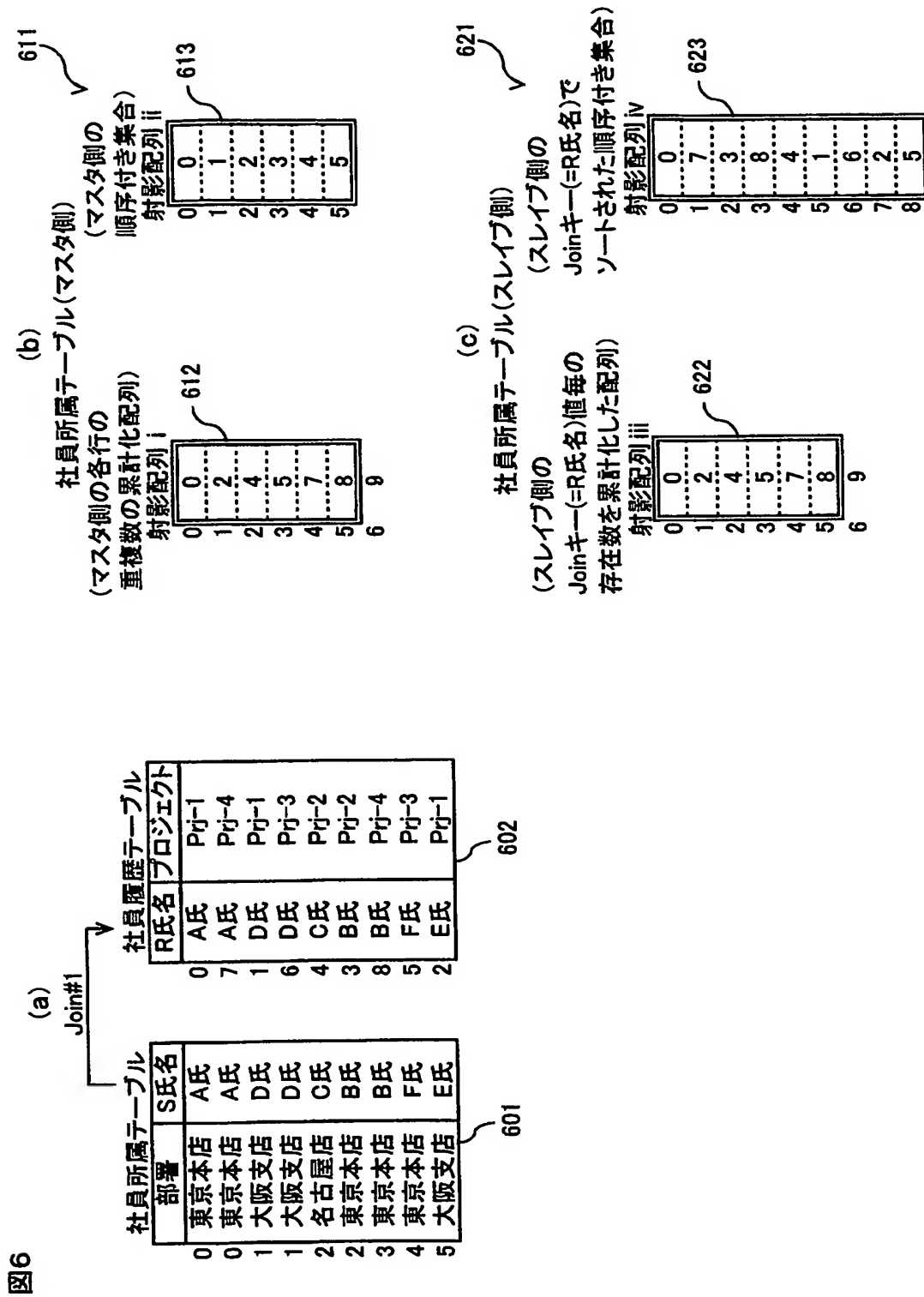
図4



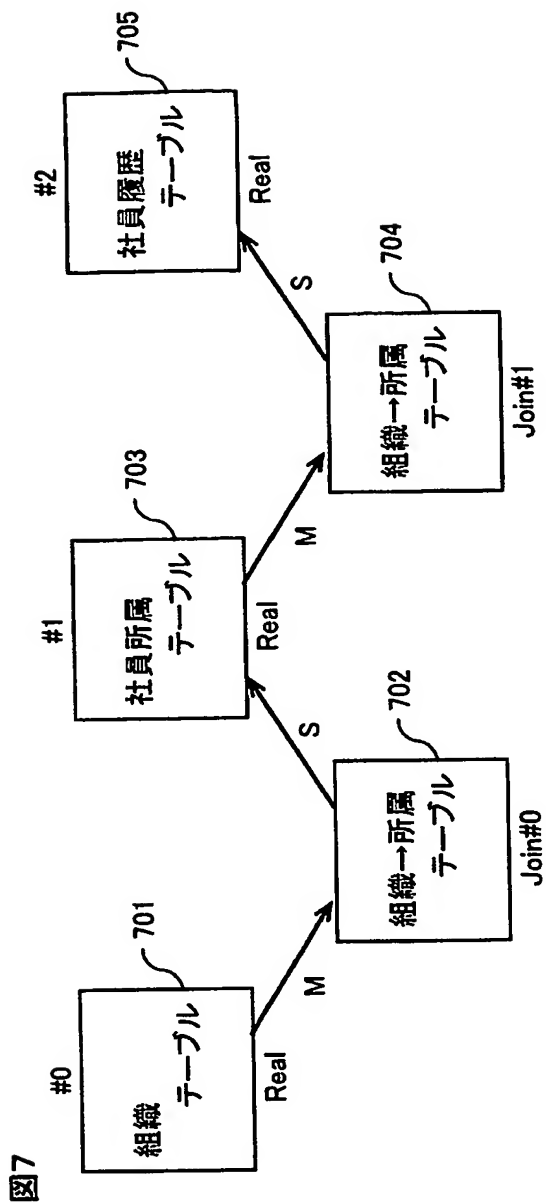
【図 5】



【図 6】

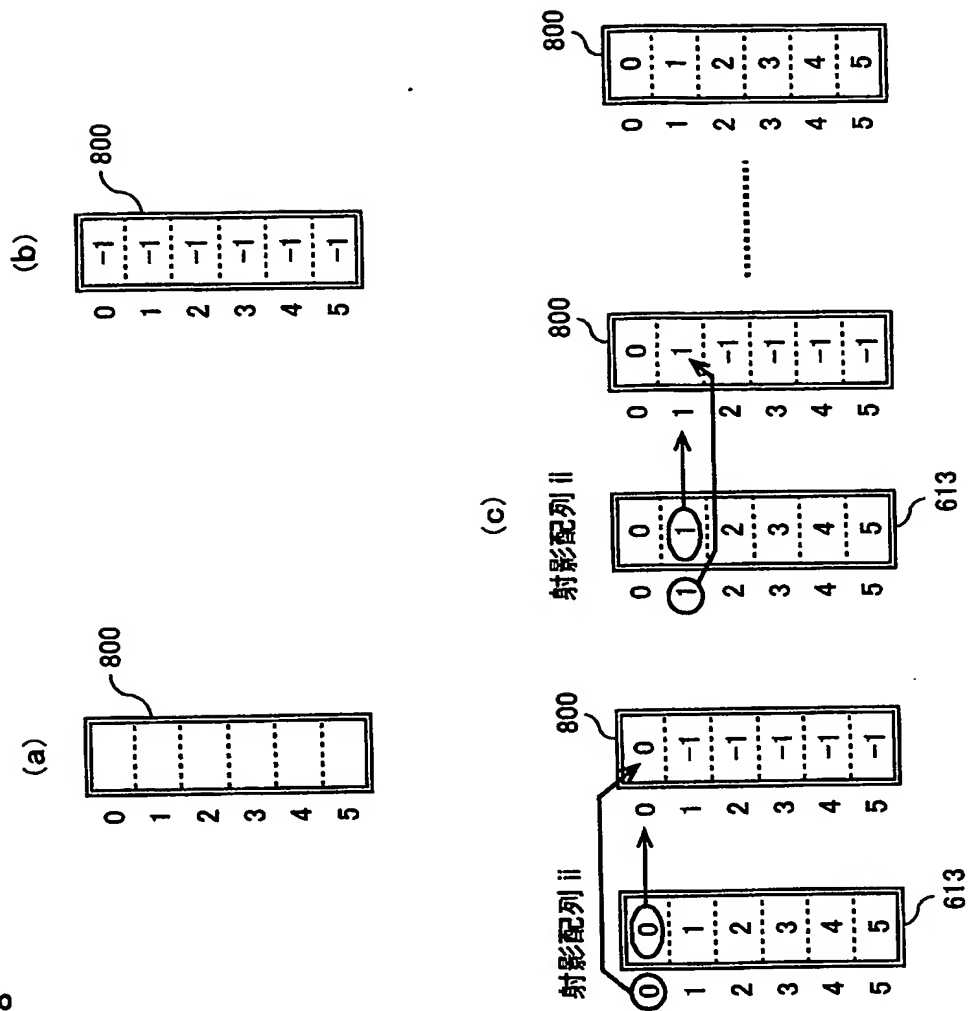


【図 7】

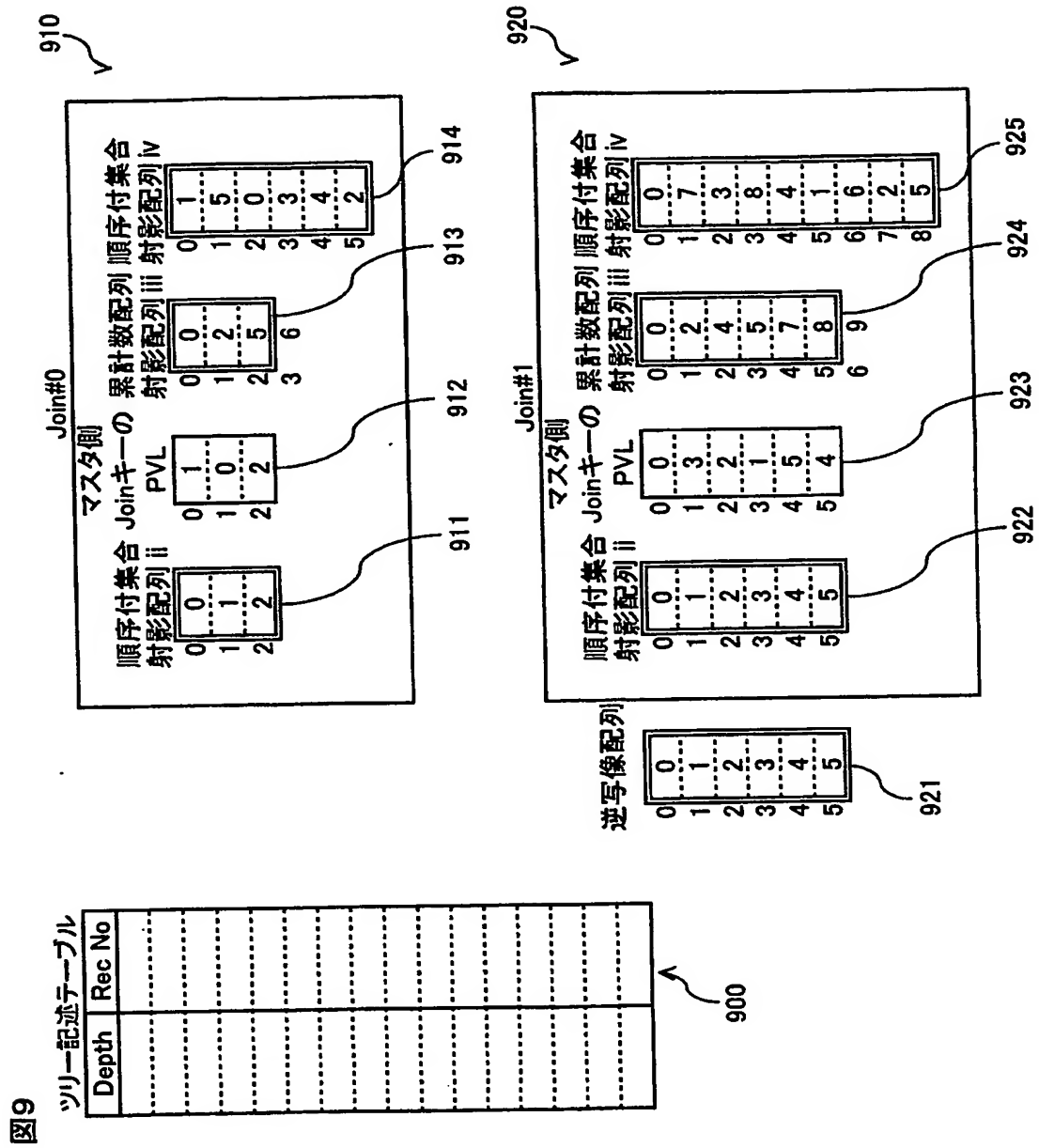


【図 8】

図 8



【圖 9】



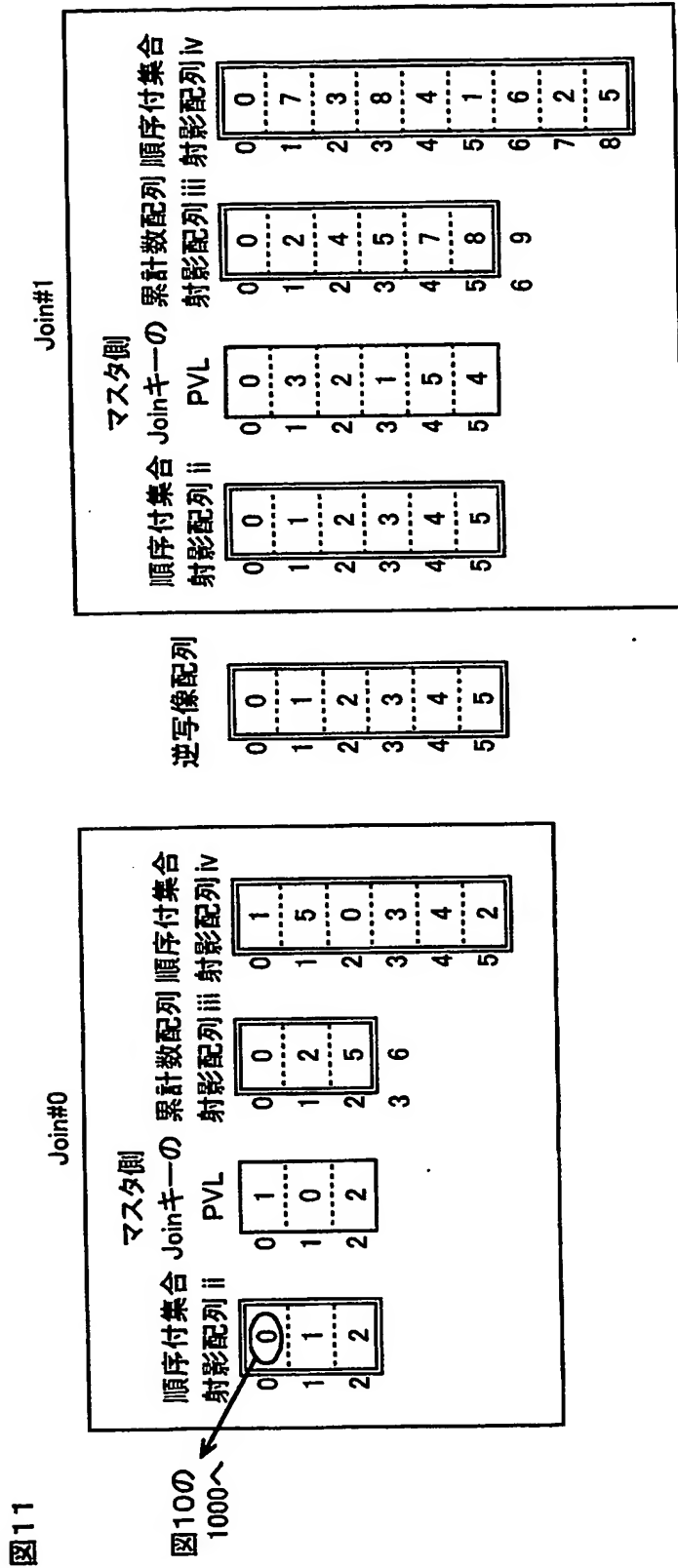
【図10】

図10

900
✓

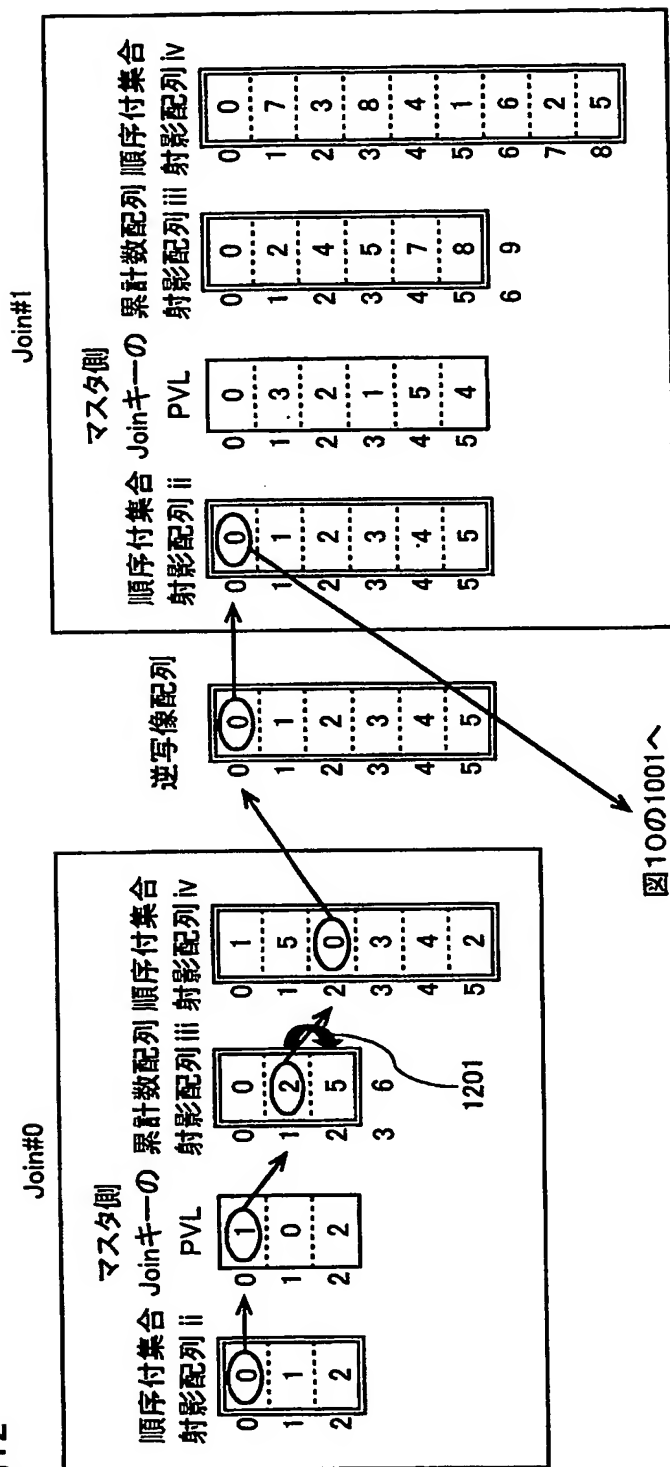
Depth	Rec No	
0	0	← 1000
1	0	← 1001
2	0	← 1002
2	7	← 1003
1	3	← 1004
2	3	← 1005
2	8	← 1006
1	4	← 1007
2	5	← 1008
0	1	← 1009
1	1	← 1010
2	1	← 1011
2	6	← 1012
1	5	← 1013
2	2	← 1014
0	2	← 1015
1	2	← 1016
2	4	← 1017

【図 11】



【図 12】

図12



【図 13】

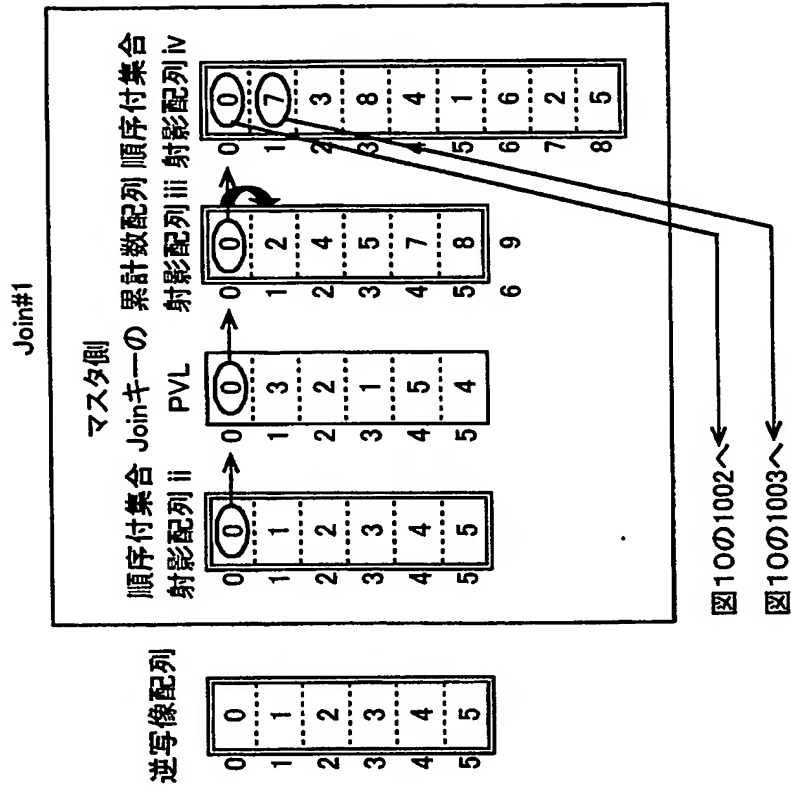
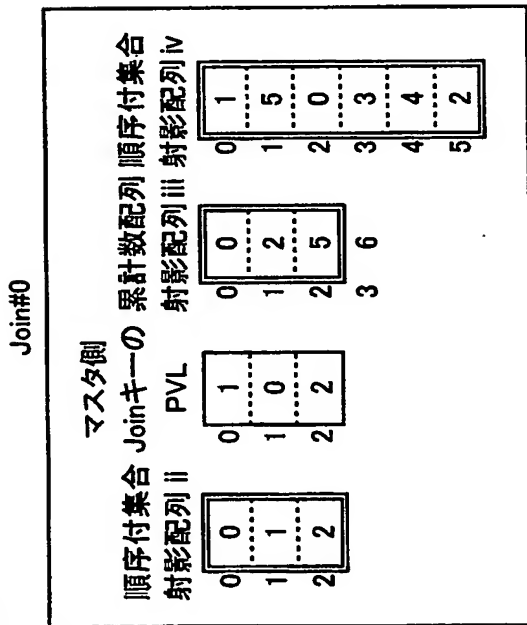
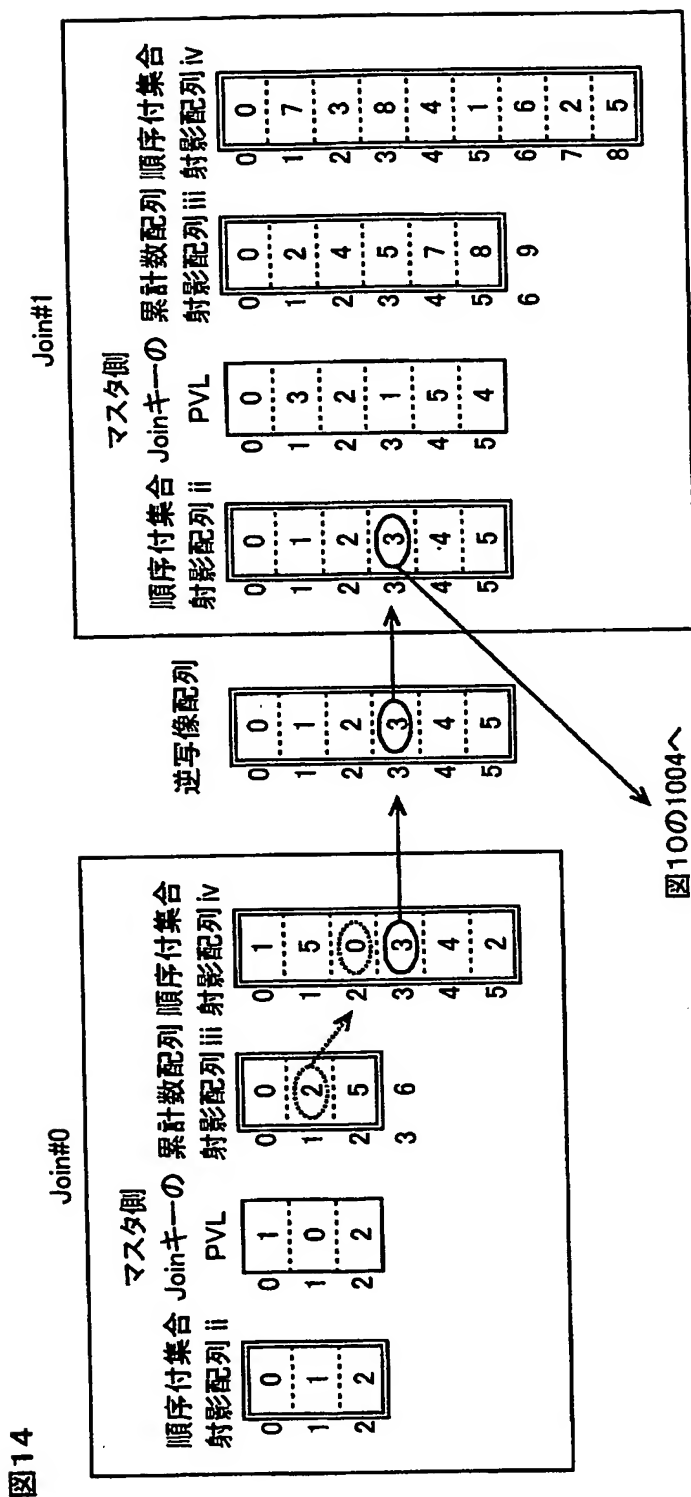


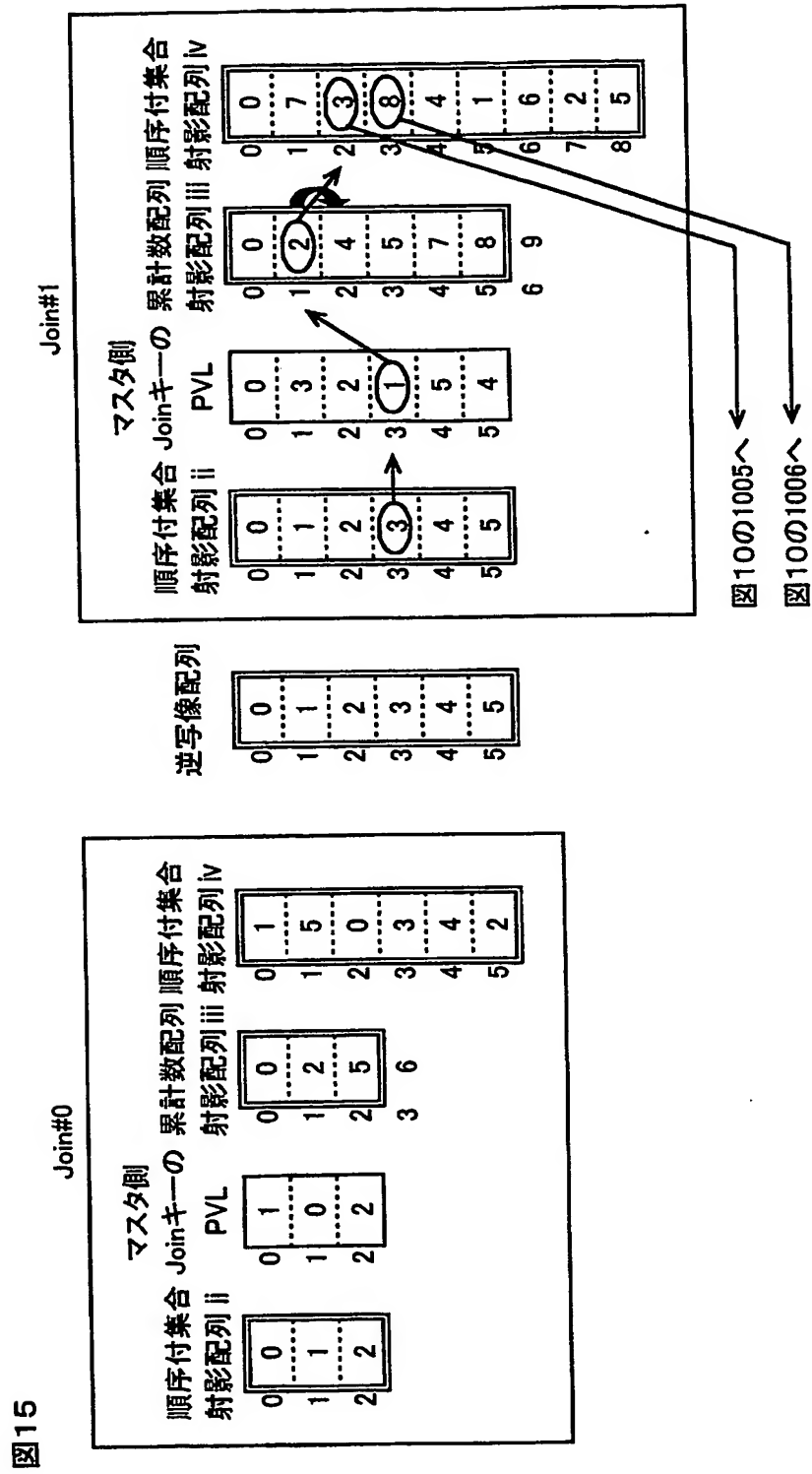
図13



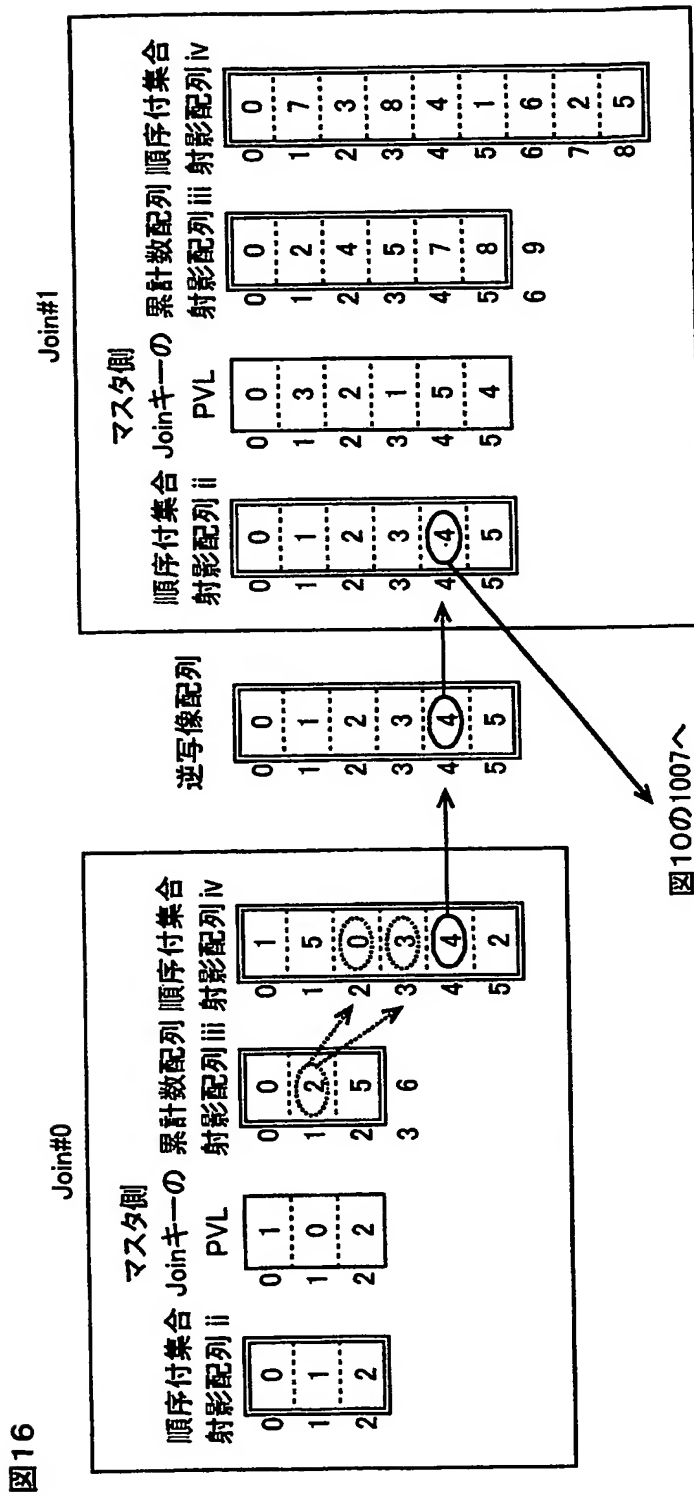
【図14】



【図15】



【図16】



【図 17】

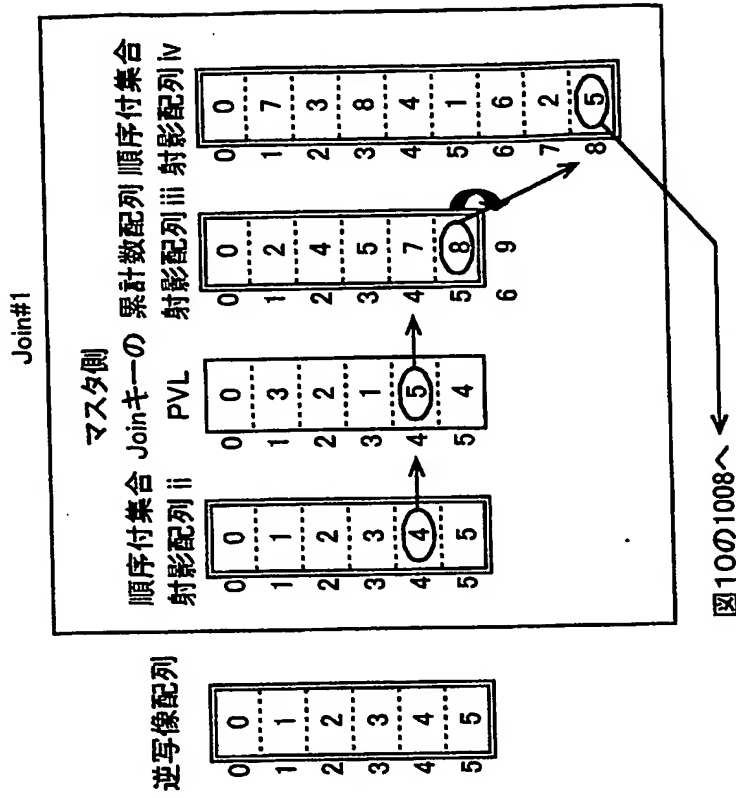
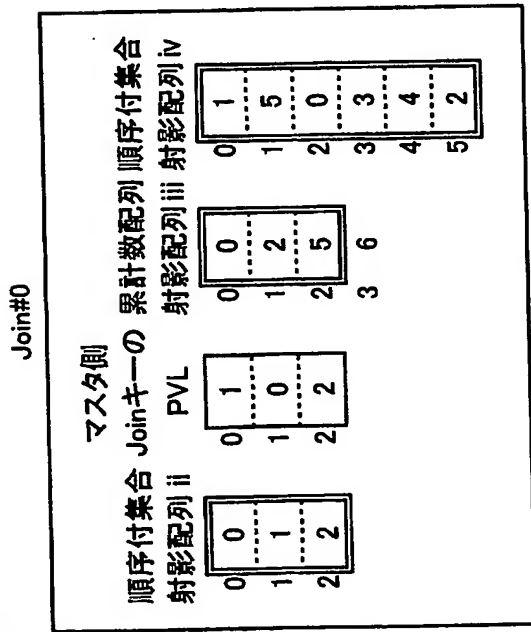


図 17



【図 18】

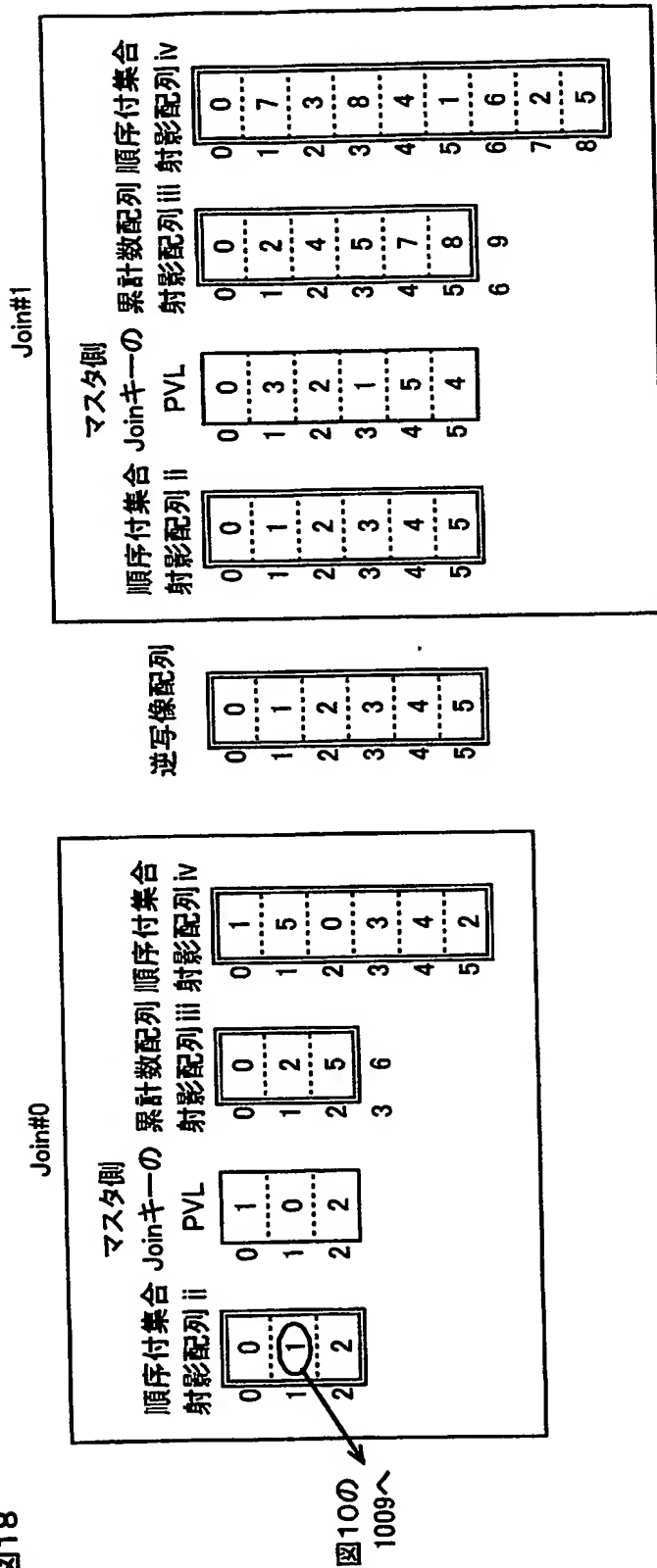


図18

【図19】

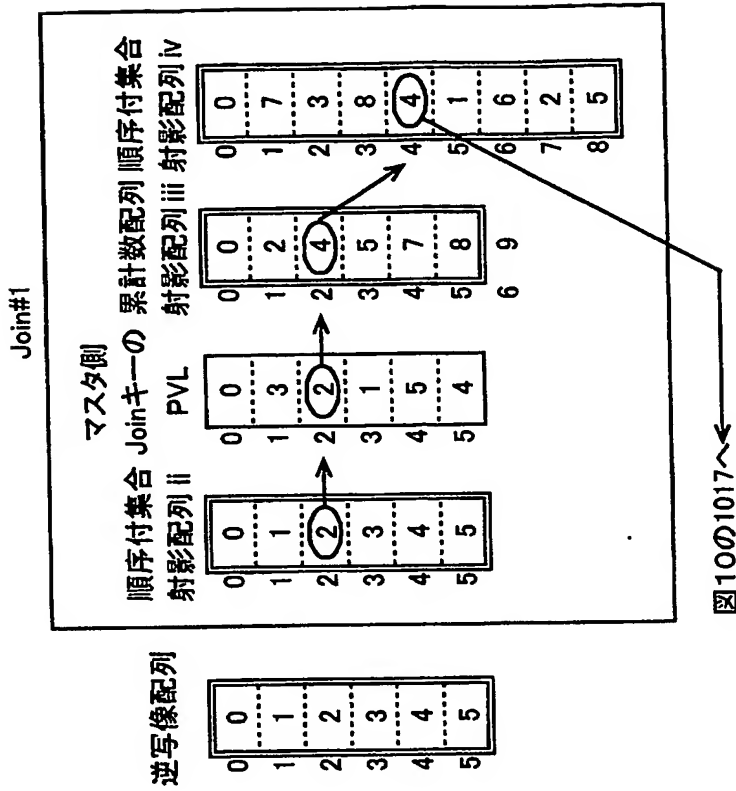
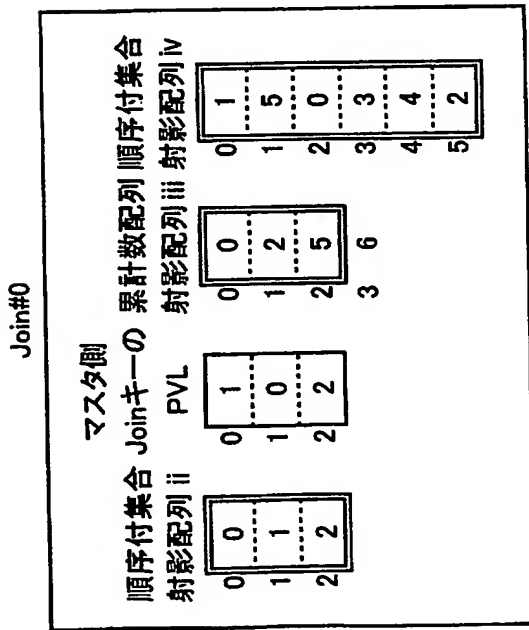
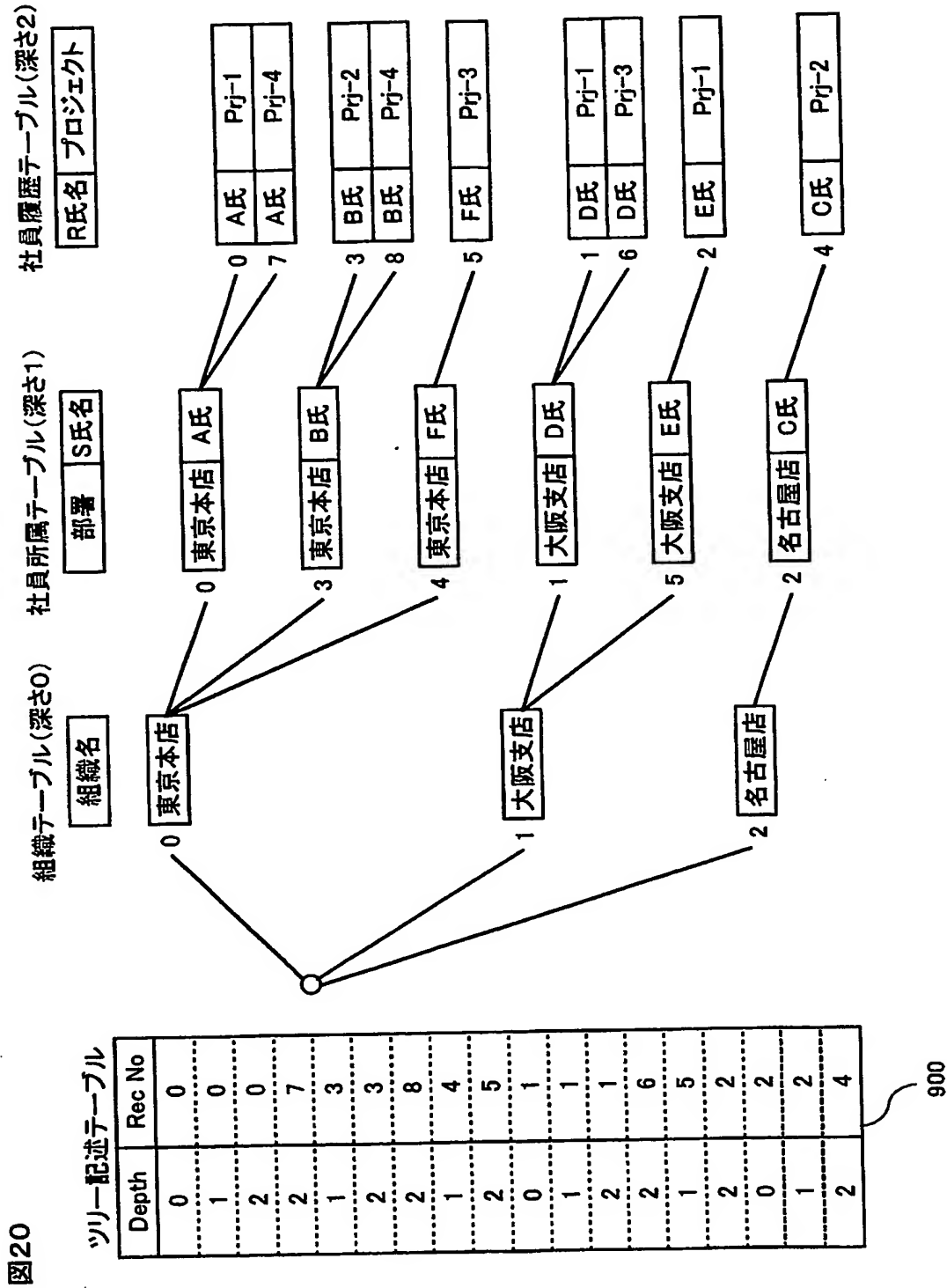


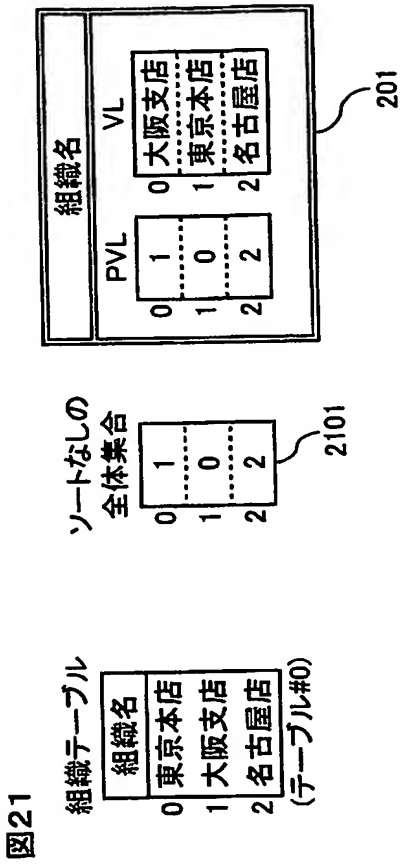
図19



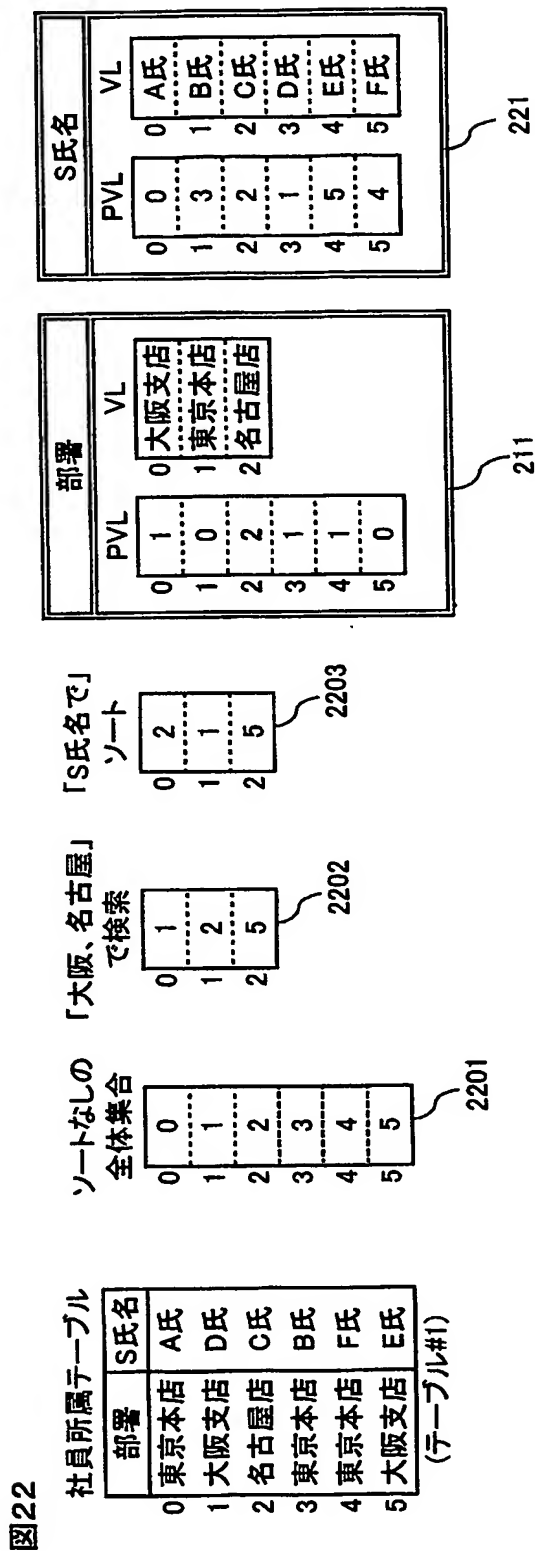
【図20】



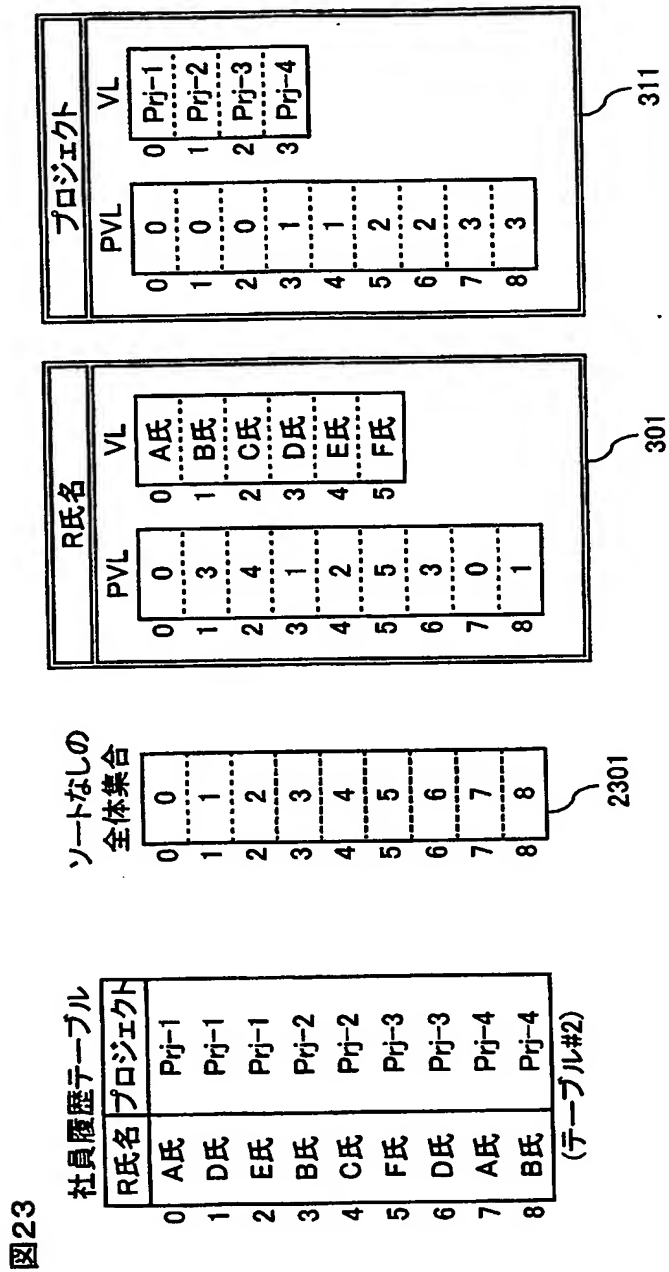
【図21】



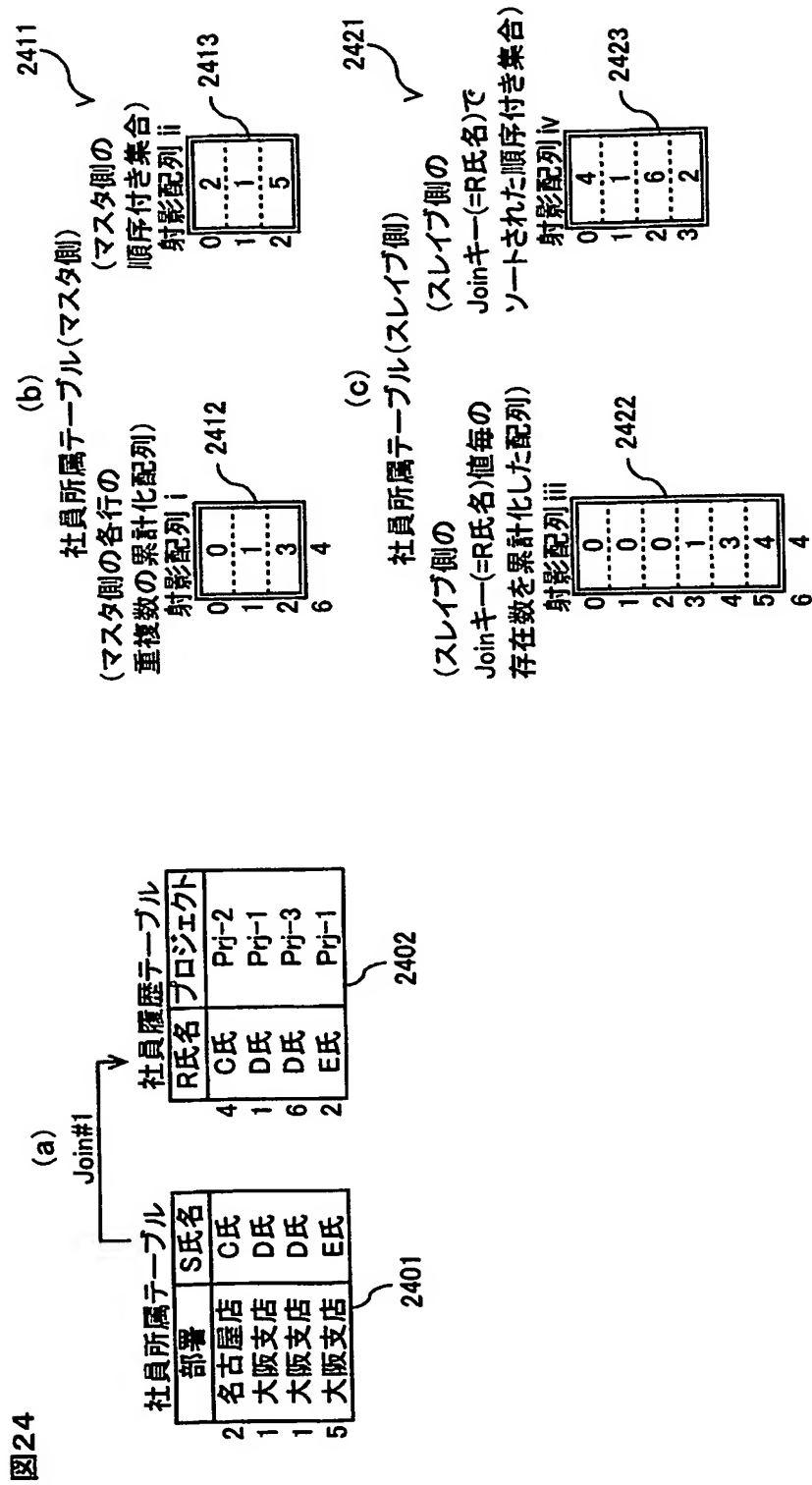
【図 22】



【図 23】

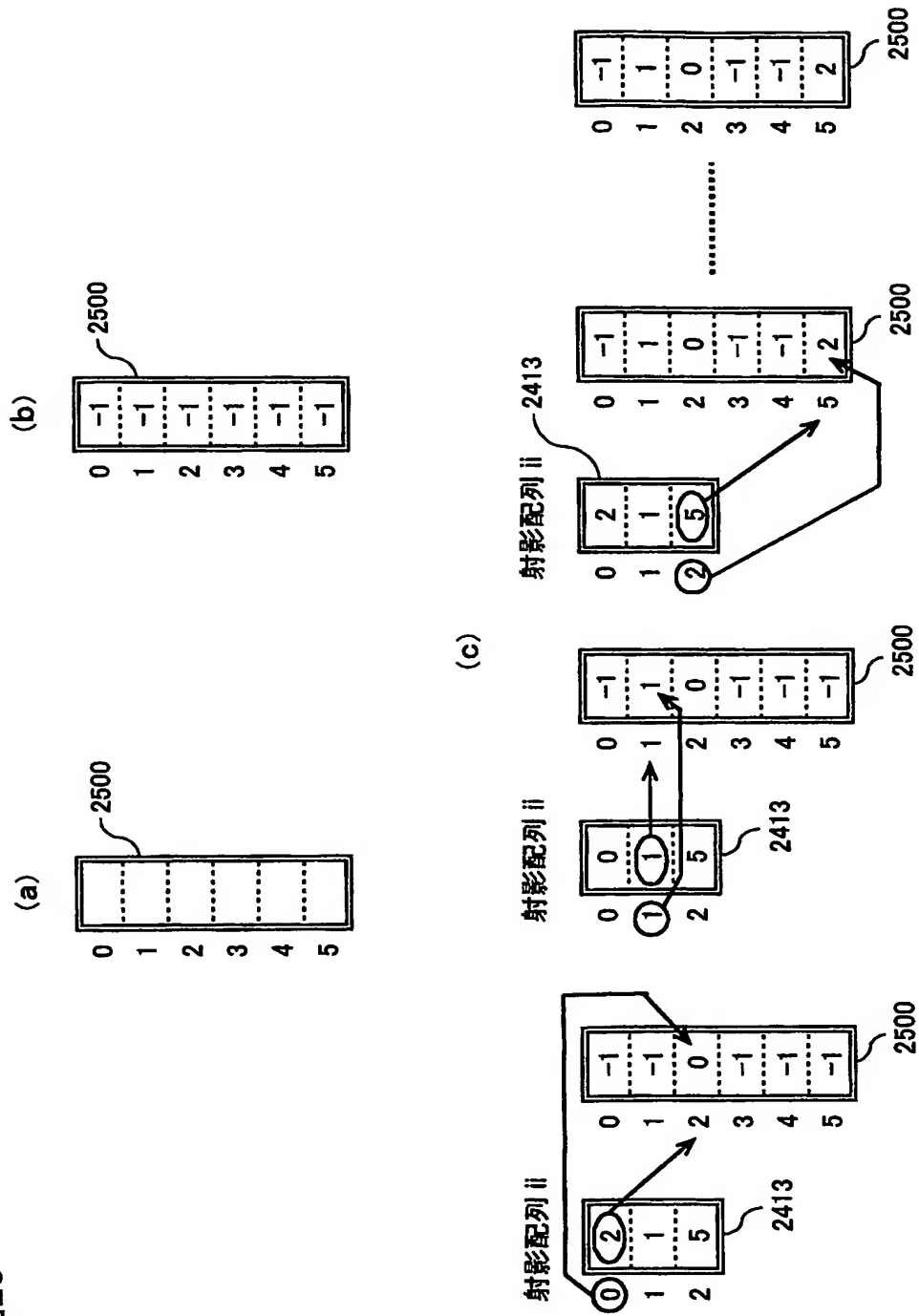


【図 2 4】

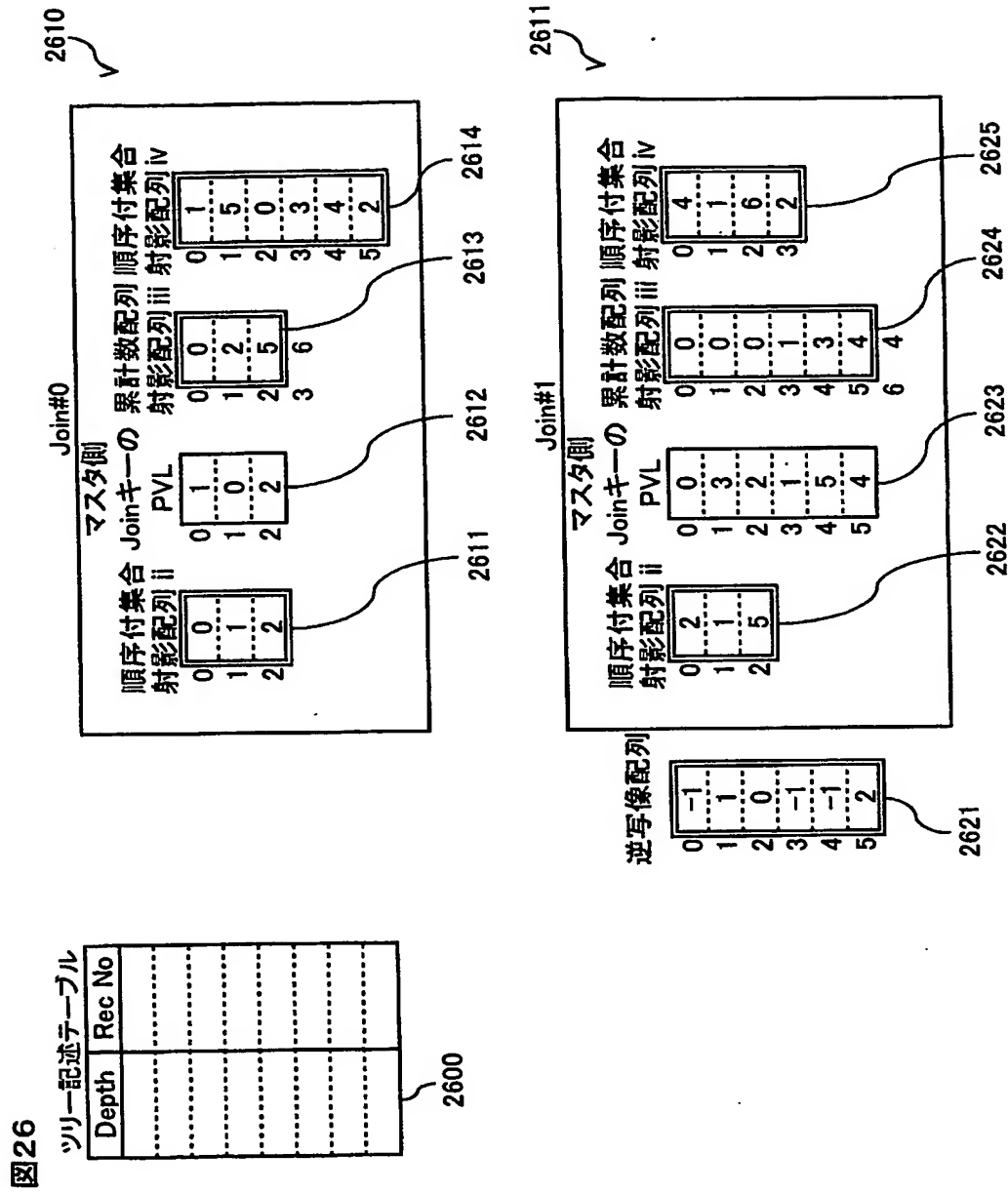


【図 2 5】

図25



【図 26】



【図 27】

図27

2600
✓

Depth	Rec No	
0	0	← 2700
0	1	← 2701
1	1	← 2702
2	1	← 2703
2	6	← 2704
1	5	← 2705
2	2	← 2706
0	2	← 2707
1	2	← 2708
2	4	← 2709

【図 28】

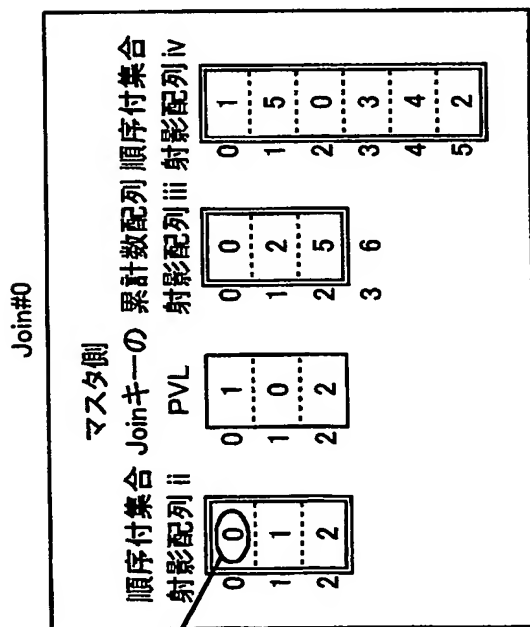
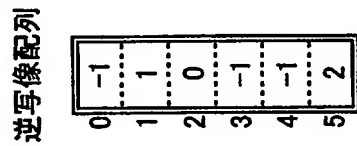
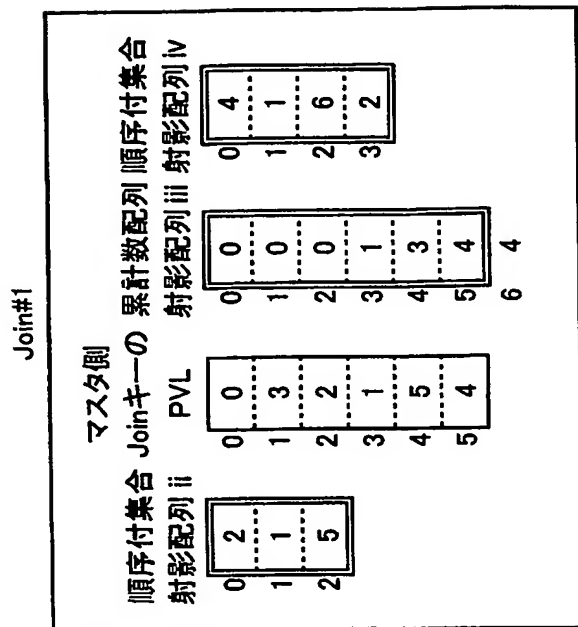
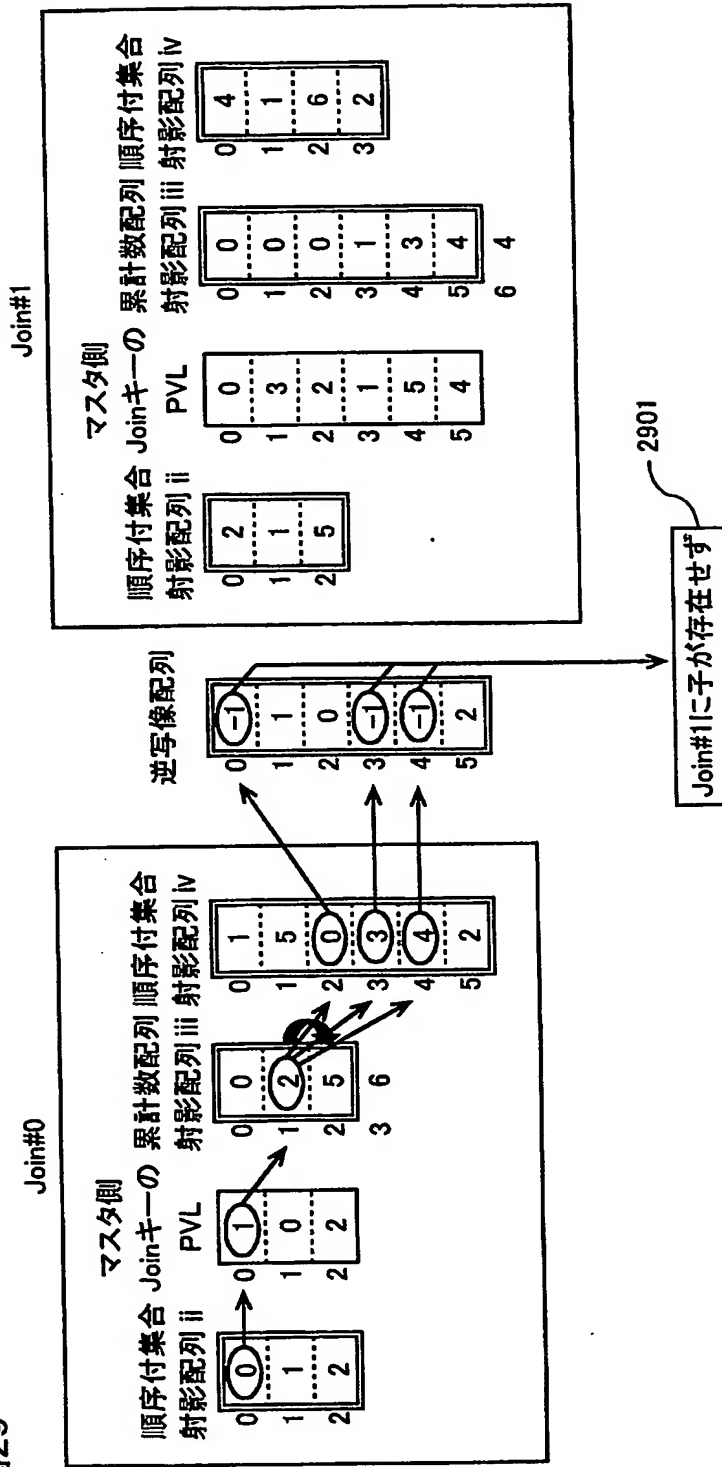


図27の
2700へ

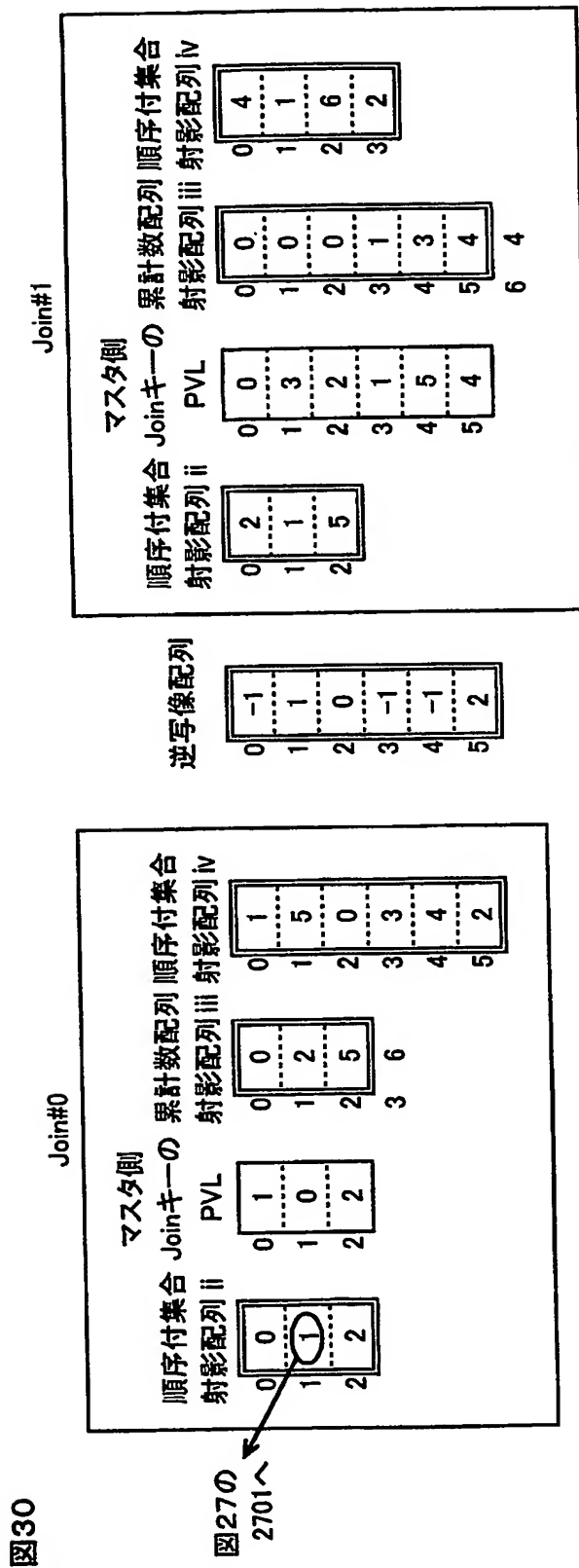
図 28

【図 29】

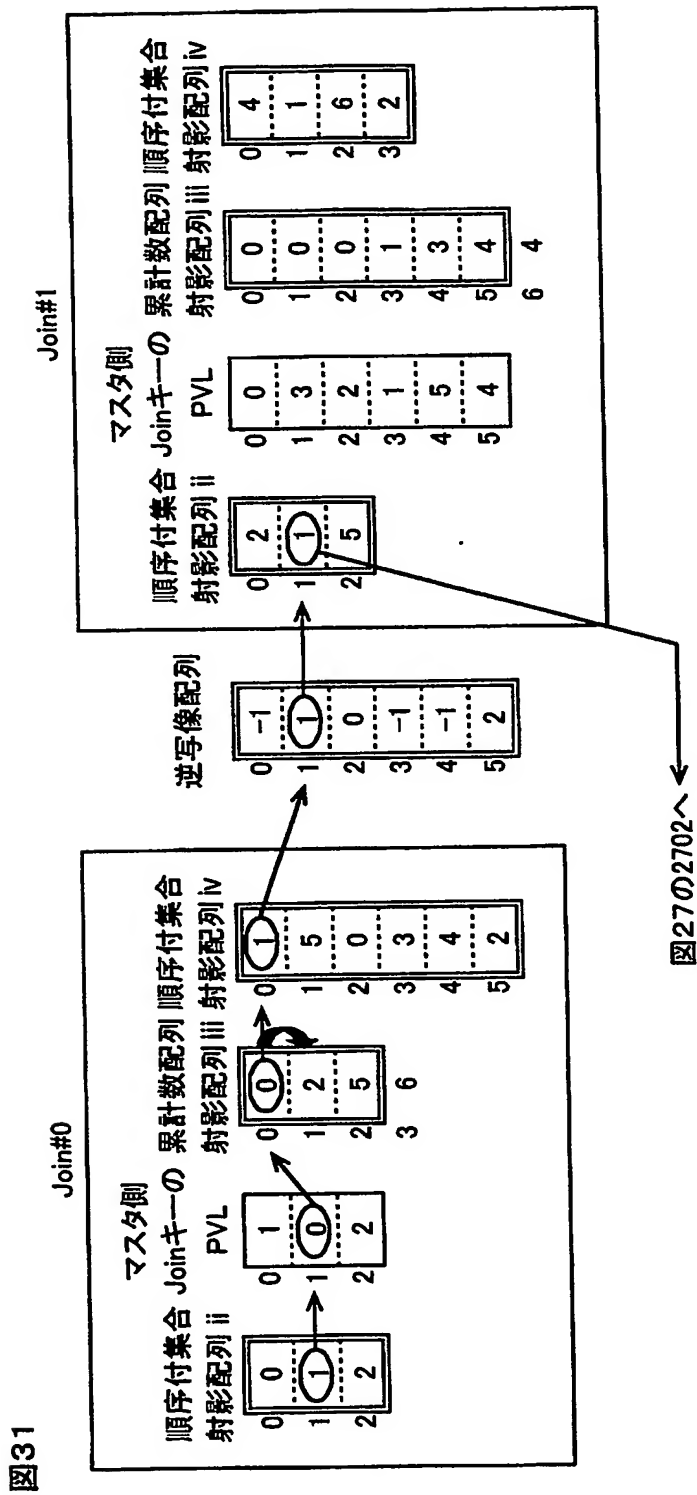
図 29



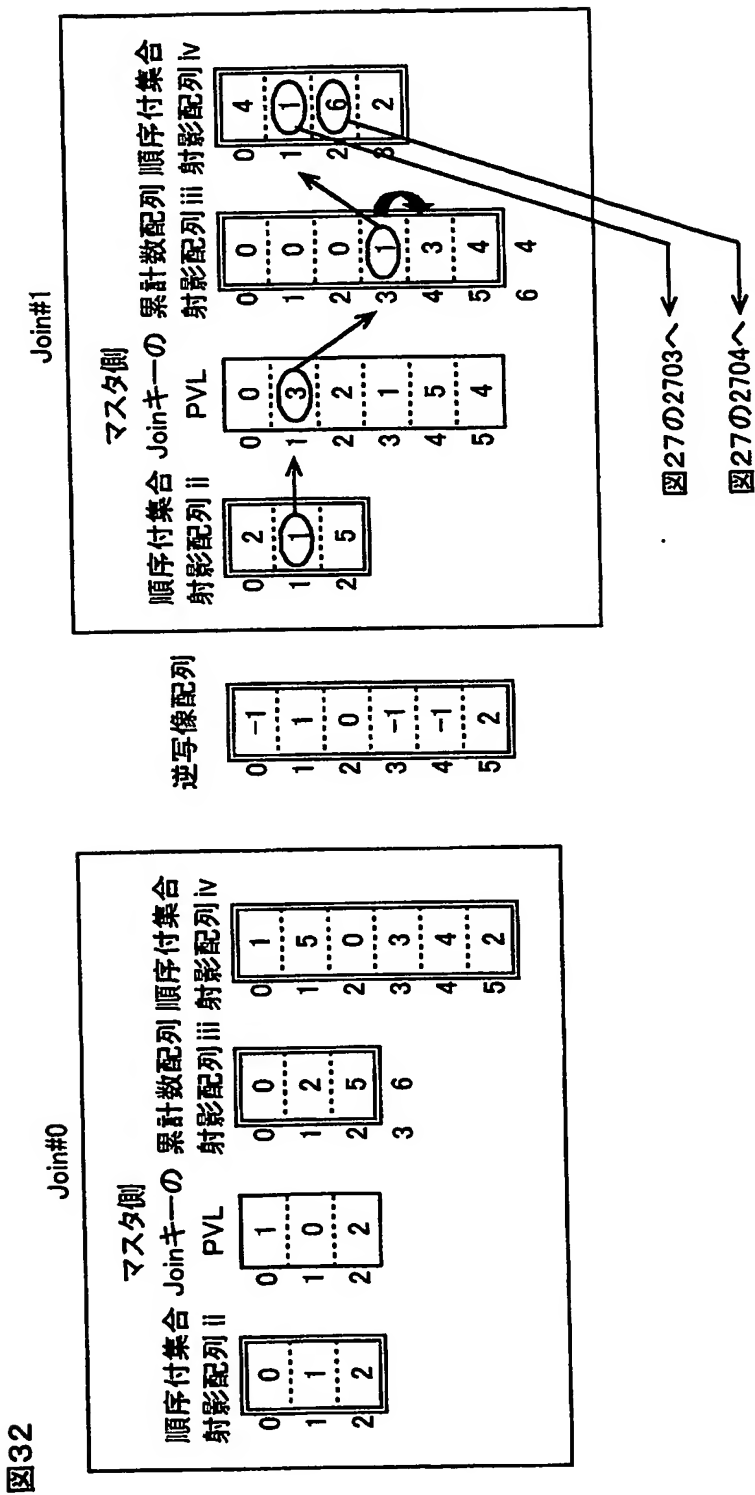
【図 30】



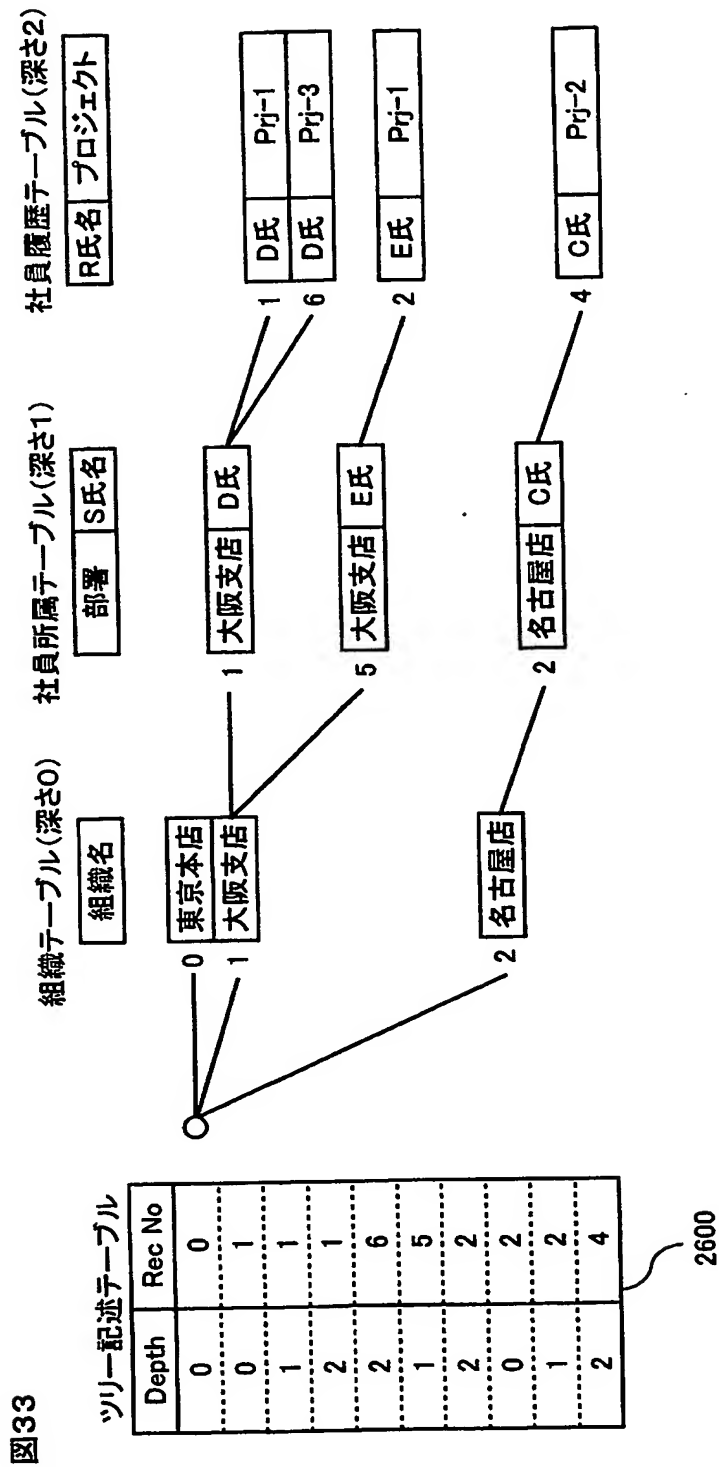
【図 31】



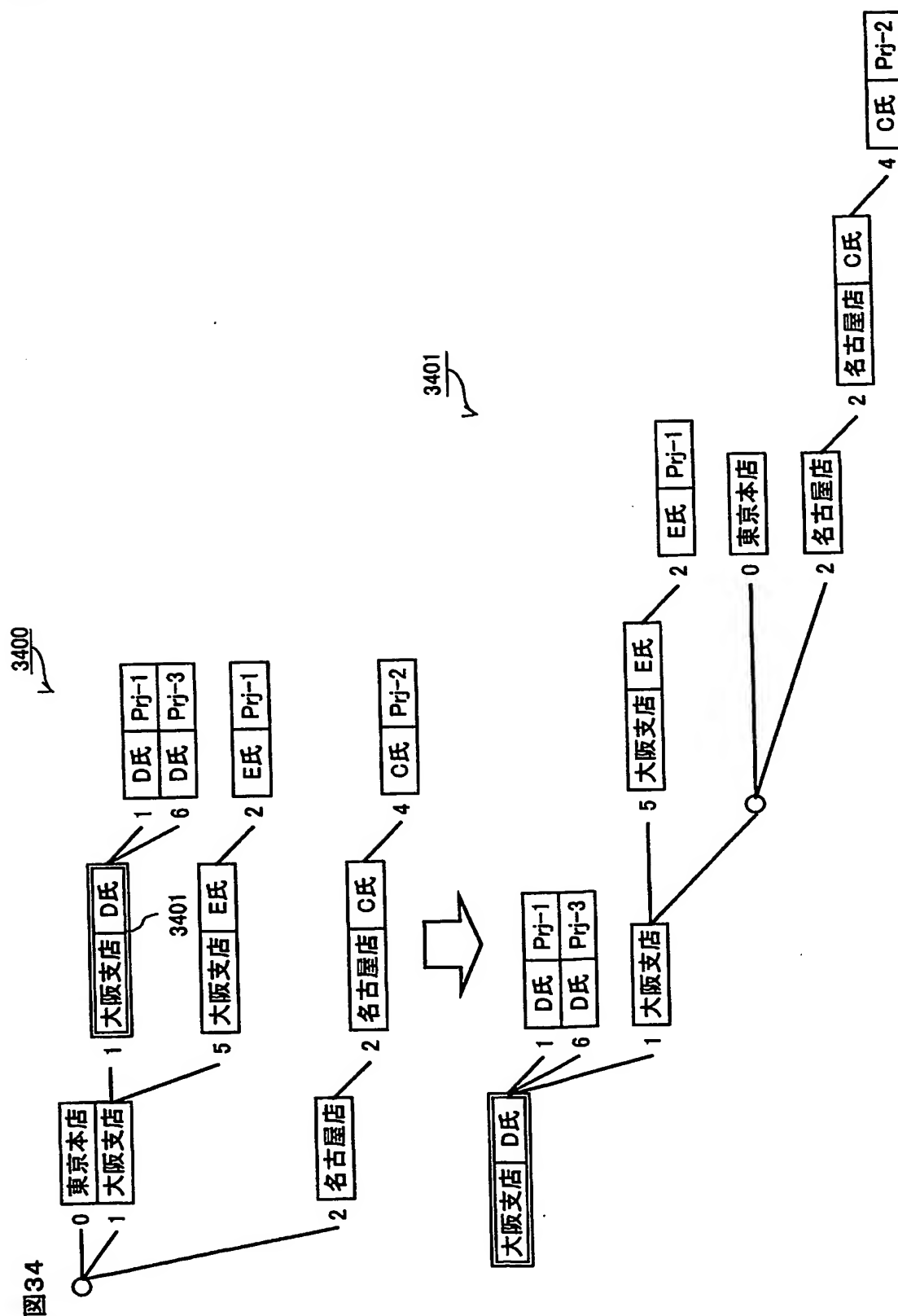
【図 3 2】



【図33】



【図 3 4】



【書類名】 要約書

【要約】

【課題】 連鎖したジョインテーブルをツリー構造に、高速変換する。

【解決手段】 まず、各表形式データを、各々が、特定の項目に属する項目値に対応した項目値番号の順に当該項目値が格納されている値リストと、一意的なレコード番号の順に、当該項目値番号を指示するためのポインタ値が格納されたポインタ配列とからなる一以上の情報ブロックに分割する。次いで、項目を共通化すべき表形式データを選択して、所定の項目の値リストを等価にして、ジョインを実行する（ステップ401）。ジョインは、必要な表形式データ間で繰り返される。次いで、結合された表形式データのうち、根となるものを選択し、表形式データの結合にしたがって、各表形式データの深さを決定し、表形式データの結合および深さにしたがって、レコードを示す値を、ツリーを記述するためのツリー記述テーブルに配置していく（ステップ404～411）。

【選択図】 図4

認定・付加情報

特許出願の番号	特願2002-309680
受付番号	50201604105
書類名	特許願
担当官	第七担当上席 0096
作成日	平成14年11月14日

<認定情報・付加情報>

【提出日】 平成14年10月24日

次頁無

特願 2 0 0 2 - 3 0 9 6 8 0

出 願 人 履 歴 情 報

識別番号

[5 0 2 3 6 9 0 1 2]

1. 変更年月日

2 0 0 2 年 9 月 5 日

[変更理由]

新規登録

住 所

神奈川県横浜市神奈川区松見町四丁目 1 1 0 1 番地 7

氏 名

株式会社ターボデータラボラトリー